Automatic Target Recognition Using Discriminative Graphical Models

Umamahesh Srinivas[†] Vishal Monga[†] Raghu G. Raj[‡]

[†]Pennsylvania State University
University Park, USA

[‡]U.S. Naval Research Laboratory
Washington DC, USA

IEEE International Conference on Image Processing

September 12, 2011

Outline

- Introduction
- Background and Review
  1. Automatic target recognition (ATR)
  2. Graphical models
- Main Contribution
  - Learning discriminative graphical models for ATR
- Experiments and Results
- Conclusions

Introduction

- View image classification as a hypothesis testing problem:

$$\begin{aligned} H_0 &: \mathbf{x} \sim f(\mathbf{x}|H_0) \\ H_1 &: \mathbf{x} \sim f(\mathbf{x}|H_1). \end{aligned}$$

Likelihood ratio test (LRT):

$$L(\mathbf{x}) := \frac{f(\mathbf{x}|H_1)}{f(\mathbf{x}|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \tau.$$

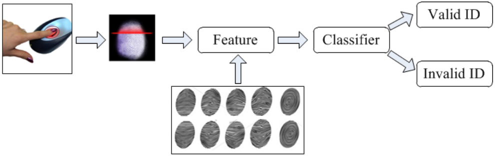


Figure: Fingerprint verification (biometrics).

- Success of Bayesian classifiers dictated by accuracy of estimation of conditional densities

Introduction

- View image classification as a hypothesis testing problem:

$$
\begin{aligned}
H_0 &: \quad \mathbf{x} \sim f(\mathbf{x}|H_0) \\
H_1 &: \quad \mathbf{x} \sim f(\mathbf{x}|H_1).
\end{aligned}
$$

Likelihood ratio test (LRT):

$$L(\mathbf{x}) := \frac{f(\mathbf{x}|H_1)}{f(\mathbf{x}|H_0)} \underset{H_0}{\overset{H_1}{\gtreqless}} \tau.$$

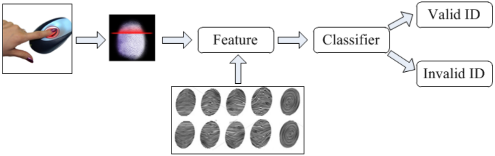


Figure: Fingerprint verification (biometrics).

- Success of Bayesian classifiers dictated by accuracy of estimation of conditional densities

Review I: Automatic Target Recognition

- Exploit imagery from diverse sensed sources for automatic target identification
- Sources: Synthetic aperture radar (SAR), inverse SAR, infra-red (FLIR), hyperspectral, etc.

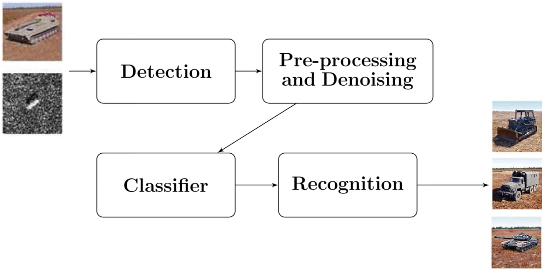


Figure: Schematic of ATR framework. The classification and recognition stages assign an input image/ feature to one of many target classes.

Target classification

Two stages in any classification framework:

1. Feature extraction from sensed imagery
2. Decision engine which performs class assignment

Algorithmic developments:

- Feature sets
  - Template-based
  - Transform domain-based (e.g. wavelets)
  - Computer vision-based
  - Estimation-theoretic
- Decision engines
  - Neural networks
  - Support vector machines (SVM)
  - Boosting
- Classifier fusion: heuristic[1], meta-classification[2,3]
  - Outputs of individual classifiers → high-level features

---

[1] Rizvi et al., Applied Imagery Pattern Recognition Workshop, 2003

[2] Sun et al., IEEE Trans. Aerosp. Electron. Syst., 2007

[3] Srinivas et al., IEEE Radar Conference, 2011

Target classification

Two stages in any classification framework:

1. Feature extraction from sensed imagery
2. Decision engine which performs class assignment

Algorithmic developments:

- Feature sets
  - Template-based
  - Transform domain-based (e.g. wavelets)
  - Computer vision-based
  - Estimation-theoretic
- Decision engines
  - Neural networks
  - Support vector machines (SVM)
  - Boosting
- Classifier fusion: heuristic[1], meta-classification[2,3]
  - Outputs of individual classifiers → high-level features

[1] Rizvi et al., Applied Imagery Pattern Recognition Workshop, 2003

[2] Sun et al., IEEE Trans. Aerosp. Electron. Syst., 2007

[3] Srinivas et al., IEEE Radar Conference, 2011

Target classification

Two stages in any classification framework:

1. Feature extraction from sensed imagery
2. Decision engine which performs class assignment

Algorithmic developments:

- Feature sets
  - Template-based
  - Transform domain-based (e.g. wavelets)
  - Computer vision-based
  - Estimation-theoretic
- Decision engines
  - Neural networks
  - Support vector machines (SVM)
  - Boosting
- Classifier fusion: heuristic[1], meta-classification[2,3]
  - Outputs of individual classifiers → high-level features

[1] Rizvi et al., Applied Imagery Pattern Recognition Workshop, 2003
[2] Sun et al., IEEE Trans. Aerosp. Electron. Syst., 2007
[3] Srinivas et al., IEEE Radar Conference, 2011

Target classification

Two stages in any classification framework:

1. Feature extraction from sensed imagery
2. Decision engine which performs class assignment

Algorithmic developments:

- Feature sets
  - Template-based
  - Transform domain-based (e.g. wavelets)
  - Computer vision-based
  - Estimation-theoretic
- Decision engines
  - Neural networks
  - Support vector machines (SVM)
  - Boosting
- Classifier fusion: heuristic[1], meta-classification[2,3]
  - Outputs of individual classifiers → high-level features

[1] Rizvi et al., Applied Imagery Pattern Recognition Workshop, 2003

[2] Sun et al., IEEE Trans. Aerosp. Electron. Syst., 2007

[3] Srinivas et al., IEEE Radar Conference, 2011

Research challenges

- Limited availability of training → serious practical concern
  - High-dimensional target image data/ equivalent features
- Variety of features and decision engines
  - No single optimal feature set-decision engine combination

Motivation for contribution:

- Presence of complementary yet correlated information
- Probabilistic graphical models: learn tractable models from high-D data under limited training.

Research challenges

- Limited availability of training → serious practical concern
  - High-dimensional target image data/ equivalent features
- Variety of features and decision engines
  - No single optimal feature set-decision engine combination

Motivation for contribution:

- Presence of complementary yet correlated information
- Probabilistic graphical models: learn tractable models from high-D data under limited training.

Review II: Graphical models

- (Undirected) 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

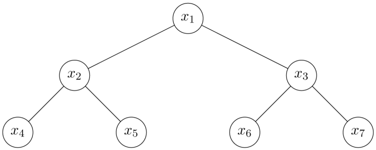


Figure: Tree - connected acyclic graph.

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

Review II: Graphical models

- (Undirected) 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

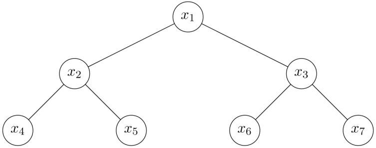


Figure: Tree - connected acyclic graph.

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

Learning graphical models

- Generative learning[4]
  - Learn a single graph to minimize approximation error:

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

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

  - Equivalent max-weight spanning tree (MWST) problem

- Discriminative learning[5]
  - Simultaneously learn a pair of graphs to minimize classification error

- Inherent trade-off:
  - Tree graphs: easy to learn, limited modeling ability
  - Learning more complex graphical structures: NP-hard

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

[5] Friedman et al., Machine Learning, 1997

Learning graphical models

- Generative learning[4]
  - Learn a single graph to minimize approximation error:

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

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

  - Equivalent max-weight spanning tree (MWST) problem
- Discriminative learning[5]
  - Simultaneously learn a pair of graphs to minimize classification error
- Inherent trade-off:
  - Tree graphs: easy to learn, limited modeling ability
  - Learning more complex graphical structures: NP-hard

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

[5] Friedman et al., Machine Learning, 1997

Learning graphical models

- Generative learning[4]
  - Learn a single graph to minimize approximation error:

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

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

  - Equivalent max-weight spanning tree (MWST) problem
- Discriminative learning[5]
  - Simultaneously learn a pair of graphs to minimize classification error
- Inherent trade-off:
  - Tree graphs: easy to learn, limited modeling ability
  - Learning more complex graphical structures: NP-hard

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

[5] Friedman et al., Machine Learning, 1997

Discriminative learning of trees[6]

Tree-approximate J-divergence of $\widehat{p}, \widehat{q}$ w.r.t. p, q:

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

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

($\widetilde{p}$ and $\widetilde{q}$: empirical distributions from $\mathcal{T}_p$ and $\mathcal{T}_q$ respectively.)

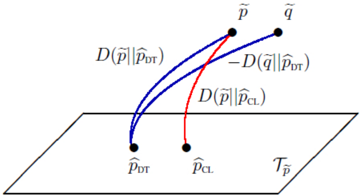


Figure: Illustration of discriminative learning (courtesy Tan et al.)

6 Tan et al., IEEE Trans. Signal Process., 2010

Discriminative vs. generative learning[7]

- Experiment: Handwritten digits classification (MNIST Database)
- Algorithms compared:
  - Chow-Liu (CL): generative learning
  - Tree Augmented Naive (TAN)
  - Discriminative Trees (DT)

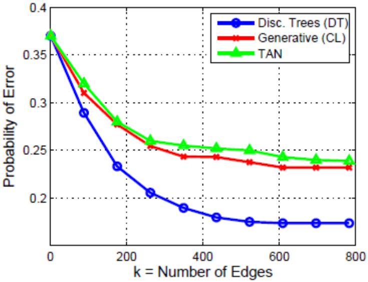


Figure: Probability of error as a function of number of newly added edges.

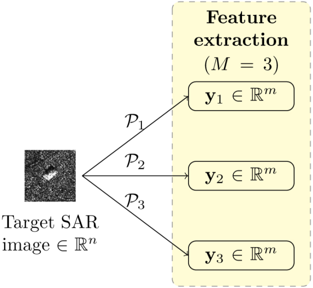
[7] Tan et al., IEEE Trans. Signal Process., 2010

Learning Discriminative Graphical Models for ATR

Two-stage framework:

1. Acquire multiple signal representations, which are conditionally correlated per class
2. Mine dependencies between different features via boosting on discriminative graphs.

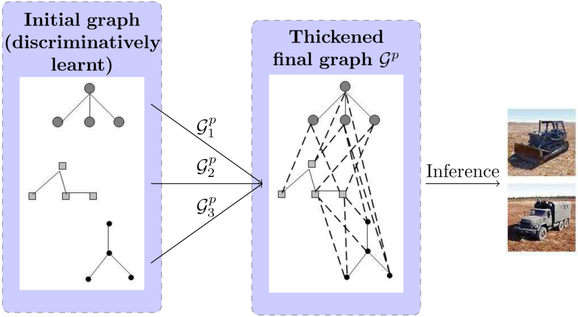
Stage 1: Feature extraction



- Projection to a lower-dimensional space $\mathcal{P} : \mathbb{R}^n \mapsto \mathbb{R}^m, m < n$
- M different projections[8] $\mathcal{P}_i, i = 1, \ldots, M$, generate corresponding low-level features $\mathbf{y}_i \in \mathbb{R}^{m_i}$

[8] For notational simplicity, we let $m_1 = m_2 = \ldots = m$.

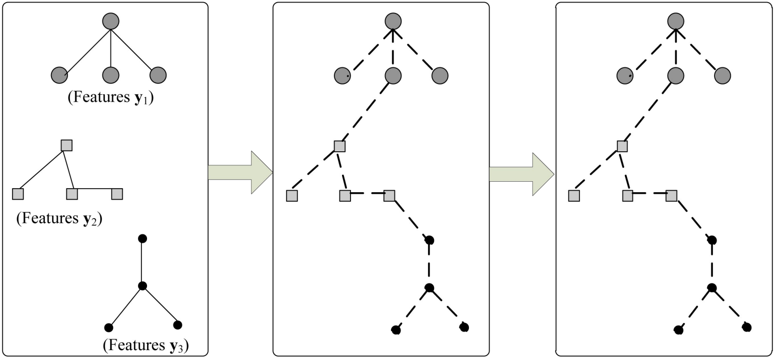
Stage 2: Learning discriminative graphs



Boosting on initially disjoint graphs to discover new edges (conditional correlations)

Learning discriminative graphs: An illustration[9]

Iteration 1:



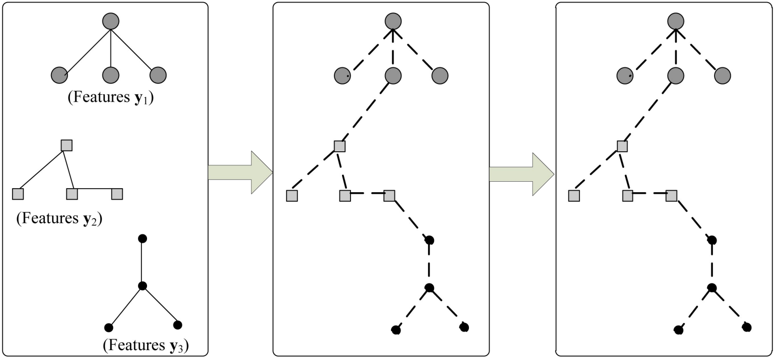
(a) Initial graph (b) Newly-learned tree (c) Augmented graph

Re-weighting of training samples (boosting) → learn another tree . . .

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

Learning discriminative graphs: An illustration[9]

Iteration 1:



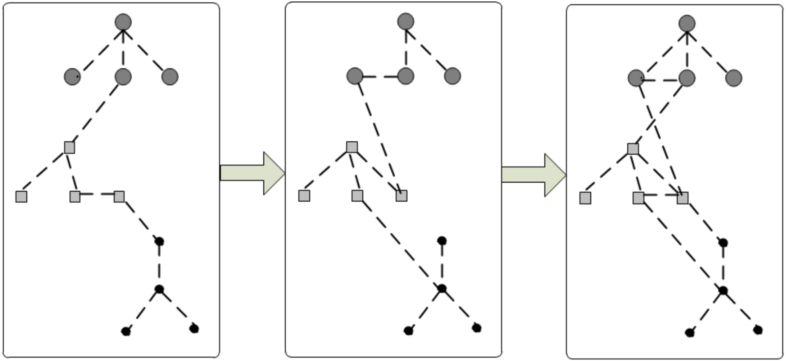
(a) Initial graph (b) Newly-learned tree (c) Augmented graph

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

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

Learning discriminative graphs: An illustration

Iteration 2:

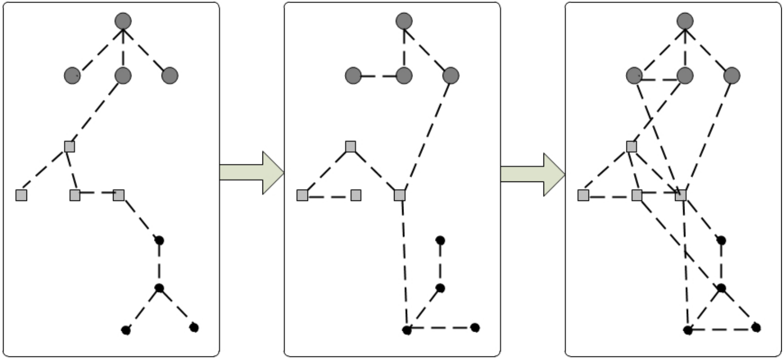


(a) Initial graph (b) Newly-learned tree (c) Augmented graph

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

Learning discriminative graphs: An illustration

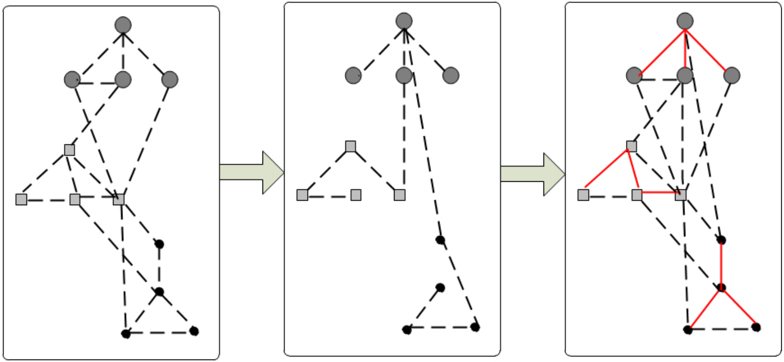
Iteration 3:



(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

How many edges to learn?

1. Cross-validation

2. Using the J-divergence:

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

Stopping criterion:
Stop after i boosting iterations if:

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

What about signal representations?

- Blind discriminative learning: no prior information about images
- Projection to wavelet sub-bands[10,11,12]
  - 2-D Reverse biorthogonal wavelets

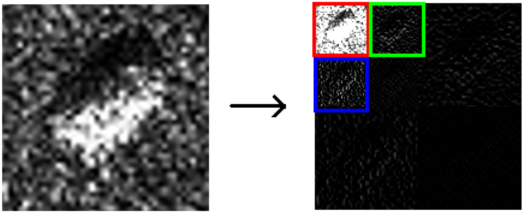


Figure: LL sub-band, LH sub-band, HL sub-band.

[10] Fukuda et al., IEEE Trans. Gesoscience and Remote Sensing, 1999

[11] Simard et al., IEEE IGARSS, 1999

[12] N. Sandirasegaram, Tech. Memo. DRDC Ottawa, 2005

Experiment: Multi-class classification for ATR[13]

Five classes from benchmark MSTAR database:

1. T-72 tanks
2. BMP-2 infantry fighting vehicles
3. BTR-70 armored personnel carriers
4. ZIL131 trucks
5. D7 tractors

- Processed input image dimension - 64×64
- Training: 150 images per class; testing: 1913 images
- Compare with single feature set + SVM.

[13] Extension of binary classification in one-versus-all manner.

Experiment: Multi-class classification for ATR[13]

Five classes from benchmark MSTAR database:

1. T-72 tanks
2. BMP-2 infantry fighting vehicles
3. BTR-70 armored personnel carriers
4. ZIL131 trucks
5. D7 tractors

- Processed input image dimension - 64×64
- Training: 150 images per class; testing: 1913 images
- Compare with single feature set + SVM.

[13] Extension of binary classification in one-versus-all manner.

Experiment: Multi-class classification for ATR

Using wavelet basis representations:

Table: Confusion matrix for LL wavelet sub-band feature + SVM.

Class	BMP-2	BTR-70	T-72	ZIL131	D7
BMP-2	**0.85**	0.04	0.04	0.03	0.04
BTR-70	0.05	**0.87**	0.03	0.02	0.03
T-72	0.04	0.07	**0.86**	0.01	0.02
ZIL131	0.01	0.05	0.06	**0.85**	0.03
D7	0.04	0.0	0.06	0.06	**0.84**

Table: Confusion matrix for proposed approach using wavelet basis.

Class	BMP-2	BTR-70	T-72	ZIL131	D7
BMP-2	**0.92**	0.05	0.02	0.01	0.01
BTR-70	0.03	**0.94**	0.02	0.0	0.01
T-72	0.02	0.05	**0.91**	0.0	0.02
ZIL131	0.01	0.02	0.03	**0.93**	0.01
D7	0.01	0.0	0.04	0.04	**0.91**

Experiment: Performance as function of training size

- Practical concern for ATR: limited training resources
- Binary classification problem: T-72 and BMP-2 classes
- Probability of misclassification $\rightarrow$ average of false-alarm and miss probabilities.
- Five approaches compared:
  1. **IndSVM**: single feature set + SVM
  2. **ClassFusion**: ranking-based classifier fusion[14]
  3. **AdaBoost**: boosting-based approach[15]
  4. **CombSVM**: concatenated feature vector + SVM
  5. **IGT**: Proposed iterative graph thickening framework

[14] Rizvi et al., Applied Imagery Pattern Recognition Workshop, 2003

[15] Sun et al., IEEE Trans. Aerosp. Electron. Syst., 2007

Experiment: Performance as function of training size

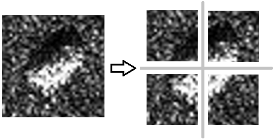
- Practical concern for ATR: limited training resources
- Binary classification problem: T-72 and BMP-2 classes
- Probability of misclassification → average of false-alarm and miss probabilities.
- Five approaches compared:
  1. **IndSVM**: single feature set + SVM
  2. **ClassFusion**: ranking-based classifier fusion[14]
  3. **AdaBoost**: boosting-based approach[15]
  4. **CombSVM**: concatenated feature vector + SVM
  5. **IGT**: Proposed iterative graph thickening framework

[14] Rizvi et al., Applied Imagery Pattern Recognition Workshop, 2003

[15] Sun et al., IEEE Trans. Aerosp. Electron. Syst., 2007

Locality-based discriminative learning

(a) Optical image.



(b) SAR image.

- Local image features more useful than global features
- Exploit scene-specific structure via image segmentation
- Wavelet LL sub-band from each region as feature.

Results: Wavelet basis

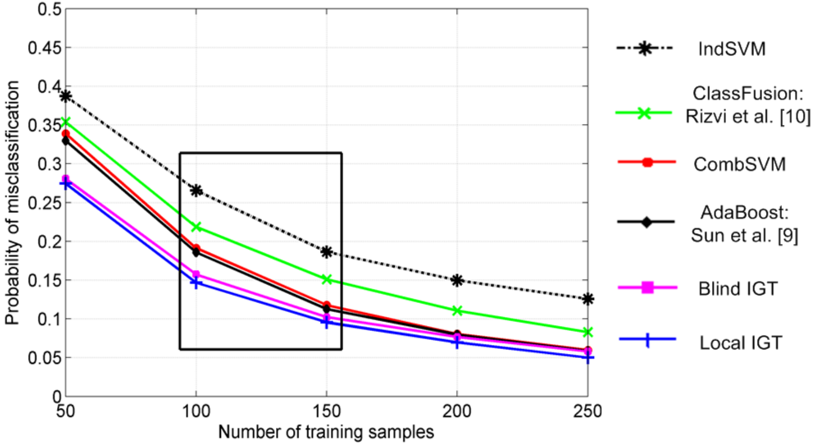


Figure: Classification error vs. training sample size. Individual feature dimension $m = 64$ (except for the local IGT method).

Conclusions

- Developed a framework to mine conditional dependencies between distinct sets of features from SAR images
  - Distinct, complementary sets of low-level features combined to exploit correlated information
    (Extension to adaptively-learnt sparse feature sets in journal version)
  - Sub-optimal discriminative graphs learnt are particularly meritorious in the difficult regime of low training, high dimensionality.

Thank you

Questions?

Backup Slides

J-divergence

Given distributions p and q,

$$J(p,q) := D(p||q) + D(q||p) = \int_{\Omega \subset \mathcal{X}^n} (p(\mathbf{x}) - q(\mathbf{x})) \log \left(\frac{p(\mathbf{x})}{q(\mathbf{x})} \right) d\mathbf{x}.$$

- Measures "separation" between tree-structured approximations $\hat{p}$ and $\hat{q}$ to arbitrary distributions p and q.

$$\frac{1}{4}\exp(-J) \leq \ \mathrm{Pr}(\mathrm{err}) \leq \frac{1}{2}\left(\frac{J}{4}\right)^{-\frac{1}{4}}.$$

- Maximize J to minimize upper bound on Pr(err).

Edge weights:

$$\begin{aligned}\psi^p_{i,j} &:= \mathbb{E}_{\widetilde{p}_{i,j}}\left[\log \frac{\widetilde{p}_{i,j}}{\widetilde{p}_i \widetilde{p}_j}\right] - \mathbb{E}_{\widetilde{q}_{i,j}}\left[\log \frac{\widetilde{p}_{i,j}}{\widetilde{p}_i \widetilde{p}_j}\right]\\ \psi^q_{i,j} &:= \mathbb{E}_{\widetilde{q}_{i,j}}\left[\log \frac{\widetilde{q}_{i,j}}{\widetilde{q}_i \widetilde{q}_j}\right] - \mathbb{E}_{\widetilde{p}_{i,j}}\left[\log \frac{\widetilde{q}_{i,j}}{\widetilde{q}_i \widetilde{q}_j}\right].\end{aligned}$$

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 $\psi^p_{i,j}$ and $\psi^q_{i,j}$ for all edges (i, j).

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

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 $H(x) =$ sign $\left[\sum_{t=1}^{T} \beta_t h_t(x)\right]$.

- Iteratively improves performance of weak learners
- 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:

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

Define:

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

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

$\rightarrow$ Thicker graphical models learnt.

ATR using sparse signal representations

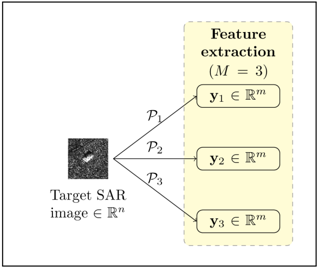
Algorithm 3 Sparse feature extraction

Given: Matrix $\mathbf{X} \in \mathbb{R}^{n \times N}$ of training vectors.

1: Dictionary learning: Adaptively learn dictionary $\mathbf{A} \in \mathbb{R}^{n \times mM}$ via K-SVD.

2: Sub-dictionaries: Divide $\mathbf{A}$ into M distinct sub-dictionaries $\mathbf{A}_i, i = 1, \ldots, M$, where $\mathbf{A}_1$ corresponds to the first m basis vectors of $\mathbf{A}$, and so on.

3: Feature: Solve M separate ℓ_1-recovery problems to obtain $\mathbf{y}_i \in \mathbb{R}^m, i = 1, \ldots, M$ corresponding to sub-dictionaries $\mathbf{A}_i$.



Here, $\mathcal{P}_i \equiv \mathbf{A}_i, i = 1, 2, 3$

ATR: Sparse signal representations

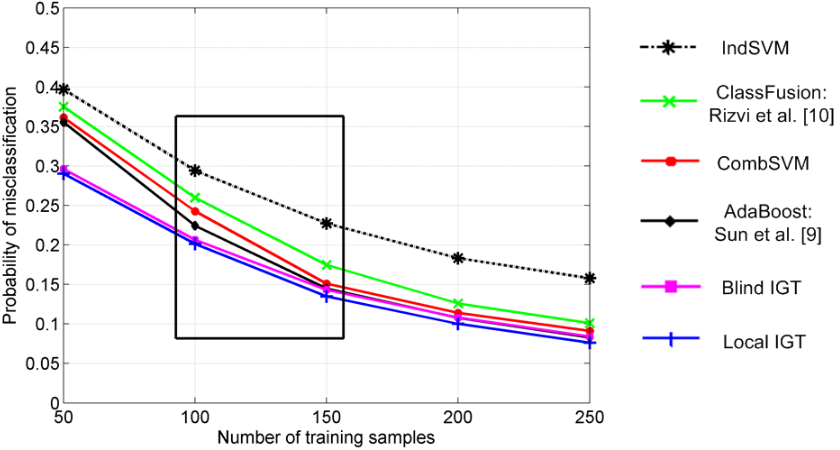


Figure: Classification error vs. training sample size. Individual feature dimension $m = 64$.

Reduced feature dimensionality: wavelet features

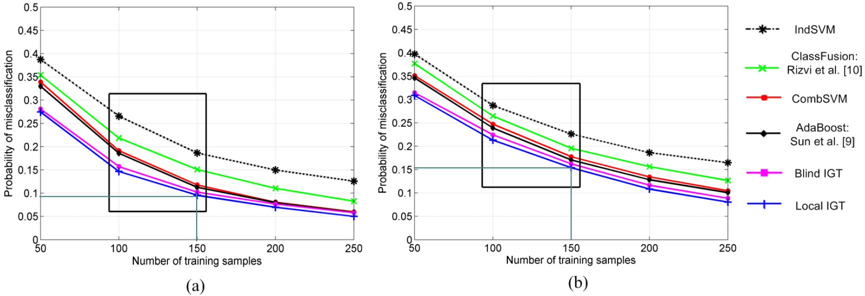


Figure: Classification error vs. training sample size. (a) Individual feature dimension $m = 64$ (except for the local IGT method). (b) Individual feature dimension $m = 16$.

Reduced feature dimensionality: sparse signal representations

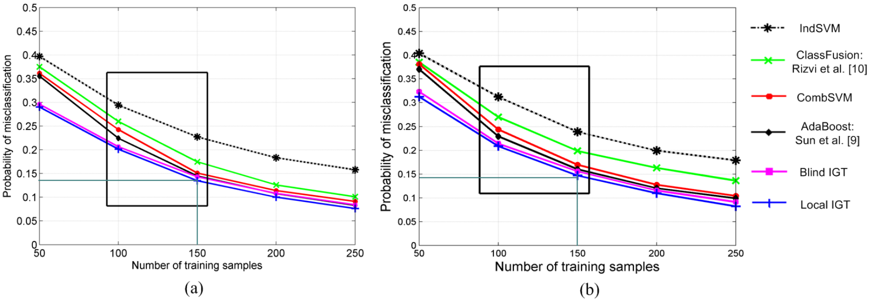


Figure: Classification error vs. training sample size. (a) Individual feature dimension $m = 64$ (except for the local IGT method). (b) Individual feature dimension $m = 16$.

Multi-class classification

- K classes $\Rightarrow$ K separate binary classification problems

Decision rule:

$$i^* = \arg \max_{i \in \{1,\ldots,K\}} \log \left(\frac{\hat{f}_{C_i}(\mathbf{y})}{\hat{f}_{\tilde{C}_i}(\mathbf{y})} \right),$$

where

- C_i: class i; $\tilde{C}_i$: complement of class i
- $\hat{f}_{C_i}$: final distribution learnt for C_i
- $\hat{f}_{\tilde{C}_i}$: final distribution learnt for $\tilde{C}_i$
- $\mathbf{y}$: test feature