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:

\[ H_0 : \mathbf{x} \sim f(\mathbf{x}|H_0) \]
\[ H_1 : \mathbf{x} \sim f(\mathbf{x}|H_1). \]

Likelihood ratio test (LRT):

\[ L(\mathbf{x}) := \frac{f(\mathbf{x}|H_1)}{f(\mathbf{x}|H_0)} \]
\[
\begin{array}{c}
H_1 \\
\H_0
\end{array}
\] \( \tau \).

**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:

\[ H_0 : \ x \sim f(x|H_0) \]
\[ H_1 : \ x \sim f(x|H_1) \]

Likelihood ratio test (LRT):

\[ L(x) := \frac{f(x|H_1)}{f(x|H_0)} \]

\[ H_1 \overset{\text{\textit{\bf{\triangleleft}}}}{\leq} H_0 \Rightarrow \tau. \]

**Figure:** Fingerprint verification (biometrics).

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

09/12/2011 ICIP 2011
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
\(^3\) 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 $\rightarrow$ high-level features

---

\(^1\) Rizvi et al., Applied Imagery Pattern Recognition Workshop, 2003
\(^3\) 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
\(^3\) 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 \(\rightarrow\) **high-level** features

---

\(^1\) Rizvi et al., Applied Imagery Pattern Recognition Workshop, 2003
\(^3\) Srinivas et al., IEEE Radar Conference, 2011

09/12/2011 ICIP 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 $\rightarrow$ 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.
(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

$$f(x) = f(x_1)f(x_2|x_1)f(x_3|x_1)f(x_4|x_2)f(x_5|x_2)f(x_6|x_3)f(x_7|x_3).$$
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

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

**Figure**: Tree - connected acyclic graph.
Learning graphical models

- **Generative learning**
  - Learn a single graph to minimize approximation error:
    
    \[
    \text{Given } p, \text{ find } \hat{p} = \arg \min_{\hat{p}_t \text{ is a tree}} D(p||\hat{p}_t).
    \]

    \[
    D(p||\hat{p}_t) := \int p(x) \log \left( \frac{p(x)}{\hat{p}_t(x)} \right) dx \rightarrow \text{KL-divergence.}
    \]

  - Equivalent max-weight spanning tree (MWST) problem

- **Discriminative learning**
  - 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

---

5 Friedman et al., Machine Learning, 1997
Learning graphical models

• **Generative learning**
  
  - Learn a single graph to minimize *approximation* error:
  
  $\text{Given } p, \text{ find } \hat{p} = \arg \min_{p_t \text{ is a tree}} D(p \| p_t).$
  
  $D(p \| p_t) := \int p(x) \log \left( \frac{p(x)}{p_t(x)} \right) dx \rightarrow \text{KL-divergence}.$

  - Equivalent max-weight spanning tree (MWST) problem

• **Discriminative learning**
  
  - 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

---

5 Friedman et al., Machine Learning, 1997
Learning graphical models

- **Generative learning**\(^4\)
  - Learn a single graph to minimize approximation error:

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

  \[
  D(p||p_t) := \int p(x) \log \left( \frac{p(x)}{p_t(x)} \right) dx \to \text{KL-divergence.}
  \]

- 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

---


\(^5\) Friedman et al., Machine Learning, 1997
Discriminative learning of trees

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

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

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

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

**Figure:** Illustration of discriminative learning (courtesy Tan et al.)
Discriminative vs. generative learning

- 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 \( \mathcal{P}_i, i = 1, \ldots, M \), generate corresponding low-level features \( y_i \in \mathbb{R}^{m_i} \)

\(^8\text{For notational simplicity, we let } m_1 = m_2 = \ldots = m.\)
Stage 2: Learning discriminative graphs

Boosting on initially disjoint graphs to discover new edges (conditional correlations)
Learning discriminative graphs: An illustration

Iteration 1:

(a) Initial graph

(Features $y_1$)

(Features $y_2$)

(Features $y_3$)

(b) Newly-learned tree

(c) Augmented graph

Re-weighting of training samples (boosting) $\rightarrow$ learn another tree ...
Learning discriminative graphs: An illustration

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

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:

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

Stopping criterion:
Stop after $i$ boosting iterations if:

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

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 - $64 \times 64$
- 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\textsuperscript{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 - $64 \times 64$
- **Training:** 150 images per class; **testing:** 1913 images
- Compare with single feature set + SVM.

\textsuperscript{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.

<table>
<thead>
<tr>
<th>Class</th>
<th>BMP-2</th>
<th>BTR-70</th>
<th>T-72</th>
<th>ZIL131</th>
<th>D7</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMP-2</td>
<td>0.85</td>
<td>0.04</td>
<td>0.04</td>
<td>0.03</td>
<td>0.04</td>
</tr>
<tr>
<td>BTR-70</td>
<td>0.05</td>
<td>0.87</td>
<td>0.03</td>
<td>0.02</td>
<td>0.03</td>
</tr>
<tr>
<td>T-72</td>
<td>0.04</td>
<td>0.07</td>
<td>0.86</td>
<td>0.01</td>
<td>0.02</td>
</tr>
<tr>
<td>ZIL131</td>
<td>0.01</td>
<td>0.05</td>
<td>0.06</td>
<td>0.85</td>
<td>0.03</td>
</tr>
<tr>
<td>D7</td>
<td>0.04</td>
<td>0.0</td>
<td>0.06</td>
<td>0.06</td>
<td>0.84</td>
</tr>
</tbody>
</table>

Table: Confusion matrix for proposed approach using wavelet basis.

<table>
<thead>
<tr>
<th>Class</th>
<th>BMP-2</th>
<th>BTR-70</th>
<th>T-72</th>
<th>ZIL131</th>
<th>D7</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMP-2</td>
<td>0.92</td>
<td>0.05</td>
<td>0.02</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>BTR-70</td>
<td>0.03</td>
<td>0.94</td>
<td>0.02</td>
<td>0.0</td>
<td>0.01</td>
</tr>
<tr>
<td>T-72</td>
<td>0.02</td>
<td>0.05</td>
<td>0.91</td>
<td>0.0</td>
<td>0.02</td>
</tr>
<tr>
<td>ZIL131</td>
<td>0.01</td>
<td>0.02</td>
<td>0.03</td>
<td>0.93</td>
<td>0.01</td>
</tr>
<tr>
<td>D7</td>
<td>0.01</td>
<td>0.0</td>
<td>0.04</td>
<td>0.04</td>
<td>0.91</td>
</tr>
</tbody>
</table>
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:
- IndSVM: single feature set + SVM
- ClassFusion: ranking-based classifier fusion
- AdaBoost: boosting-based approach
- CombSVM: concatenated feature vector + SVM
- IGT: Proposed iterative graph thickening framework

---

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

09/12/2011 ICIP 2011
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 $q$,

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

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

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

- Maximize $J$ to minimize upper bound on $\Pr(\text{err})$. 
Edge weights:

\[
\psi^p_{i,j} := \mathbb{E}_{\tilde{p}_{i,j}} \left[ \log \frac{\tilde{p}_{i,j}}{\tilde{p}_i \tilde{p}_j} \right] - \mathbb{E}_{\tilde{q}_{i,j}} \left[ \log \frac{\tilde{q}_{i,j}}{\tilde{q}_i \tilde{q}_j} \right],
\]

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

Algorithm 1 Discriminative trees (DT)

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

1. Estimate pairwise statistics \( \tilde{p}_{i,j}(x_i, x_j), \tilde{q}_{i,j}(x_i, x_j) \) 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 \( \mathcal{E}_{\hat{p}} = \text{MWST}(\psi^p_{i,j}) \) and \( \mathcal{E}_{\hat{q}} = \text{MWST}(\psi^q_{i,j}) \).
4. Get \( \hat{p} \) by projection of \( \tilde{p} \) onto \( \mathcal{E}_{\hat{p}} \); likewise \( \hat{q} \).
5. LRT using \( \hat{p} \) and \( \hat{q} \).
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 \(h_t : S \rightarrow \mathbb{R}\) with error \(\epsilon_t\)
   - 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 \(H(x) = \text{sign} \left[ \sum_{t=1}^{T} \beta_t h_t(x) \right]\).

- 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:

\[
H_T(x) = \text{sgn} \left[ \sum_{t=1}^{T} \alpha_t \log \left( \frac{\hat{p}_t(x)}{\hat{q}_t(x)} \right) \right] = \text{sgn} \left[ \log \prod_{t=1}^{T} \left( \frac{\hat{p}_t(x)}{\hat{q}_t(x)} \right)^{\alpha_t} \right]
\]

\[
= \text{sgn} \left[ \log \left( \frac{\prod_{t=1}^{T} \hat{p}_t(x)^{\alpha_t}}{\prod_{t=1}^{T} \hat{q}_t(x)^{\alpha_t}} \right) \right] = \text{sgn} \left[ \log \left( \frac{\hat{p}(x)}{\hat{q}(x)} \right) \right]
\]

Define:

\[
Z_p(\alpha) = Z_p(\alpha_1, \ldots, \alpha_T) = \sum_x \hat{p}(x); \quad Z_q(\alpha) = \sum_x \hat{q}(x)
\]

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

→ Thicker graphical models learnt.
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 \( \ell_1 \)-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 $m = 64$. 

09/12/2011 ICIP 2011
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 ⇒ \( K \) separate binary classification problems

Decision rule:

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

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 \( \tilde{C}_i \)
- \( y \): test feature