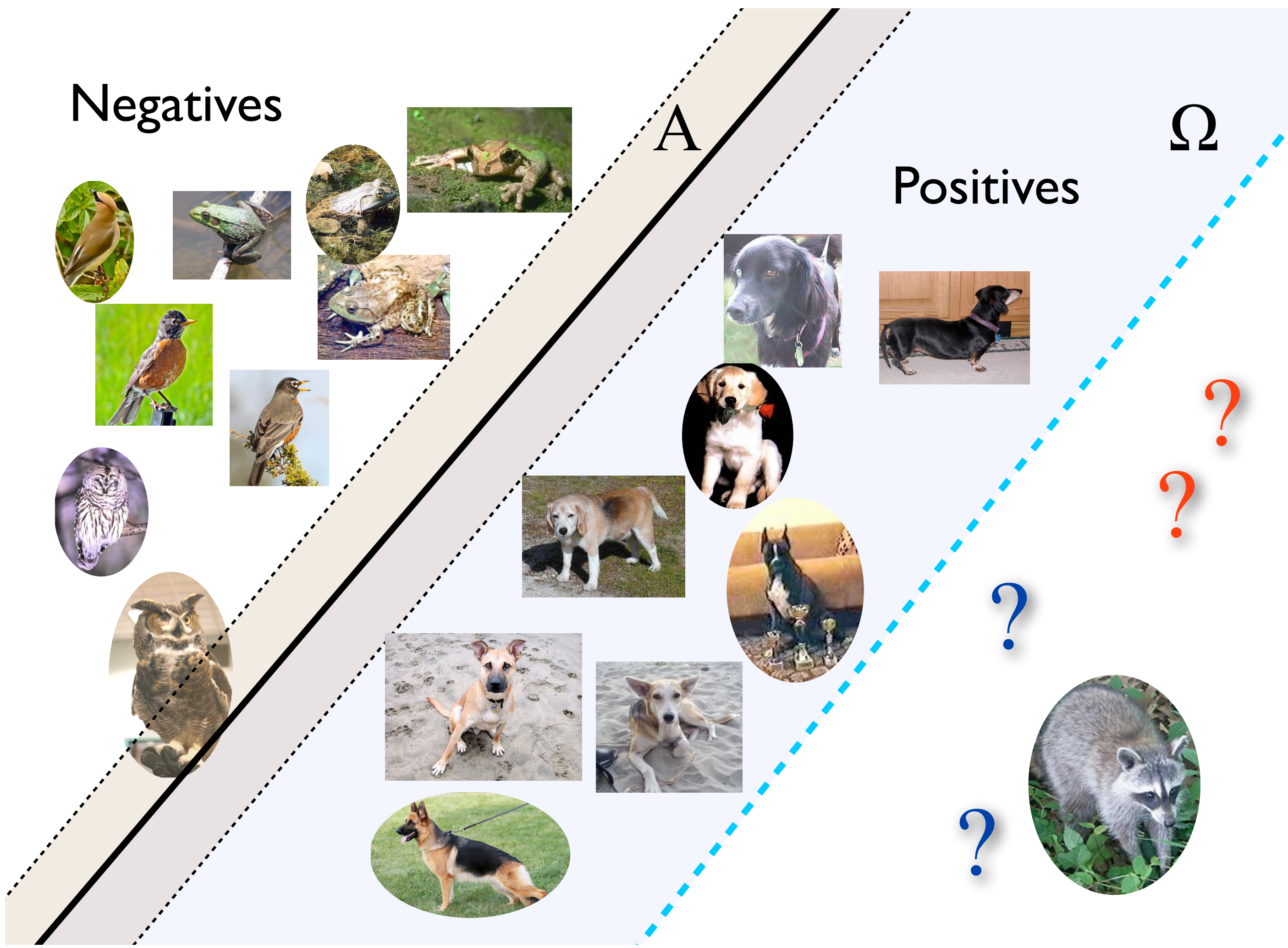
Statistical Methods for Open Set Recognition

Walter J. Scheirer and Terrance E. Boult

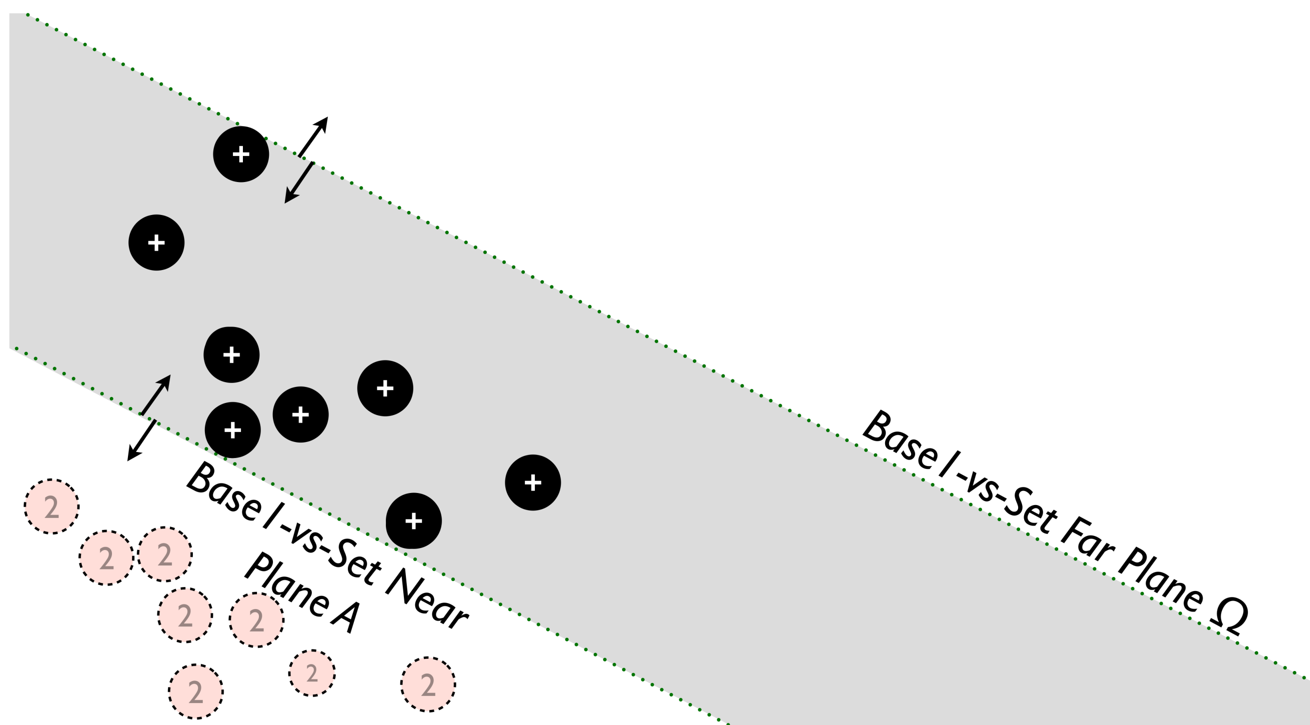
Part 3: Algorithms that Minimize the Risk of the Unknown

Let’s include open space risk in our optimization problem

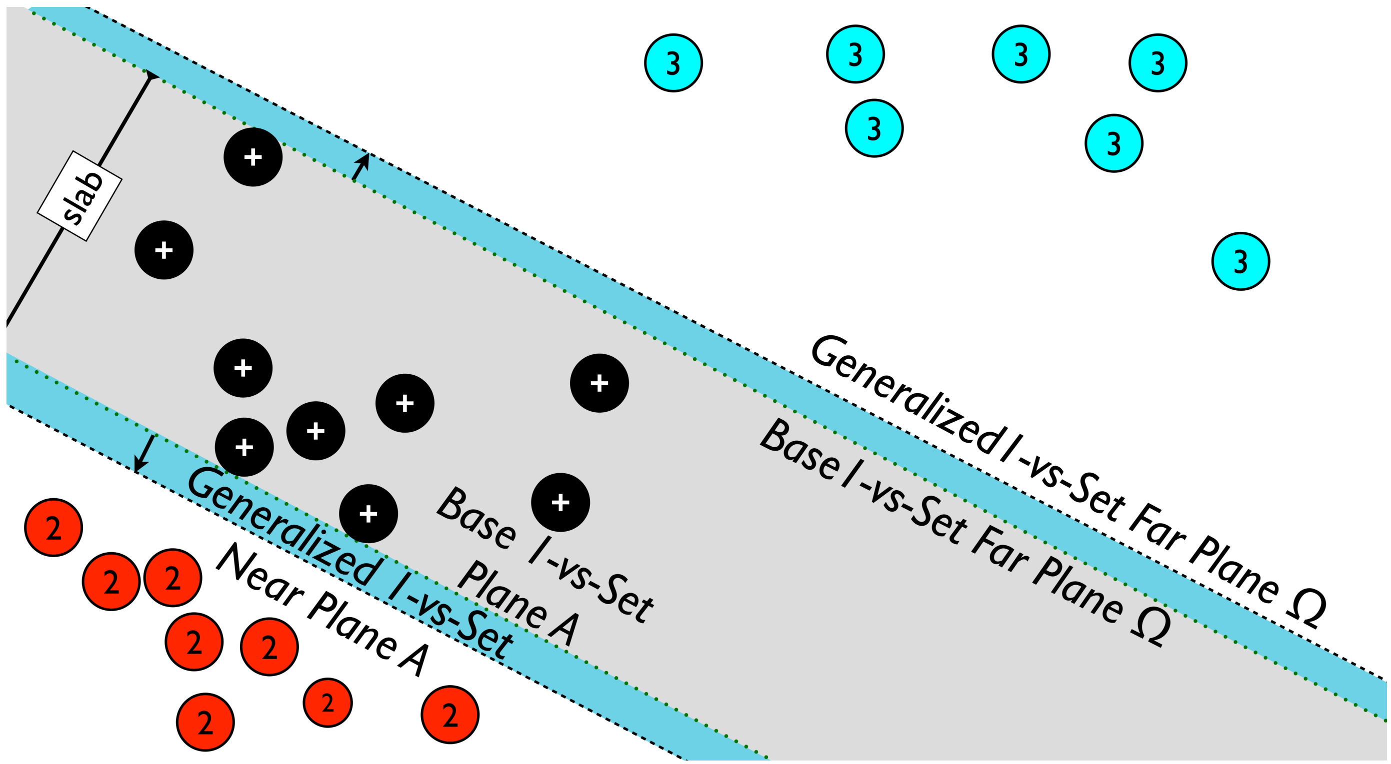
Slab Model



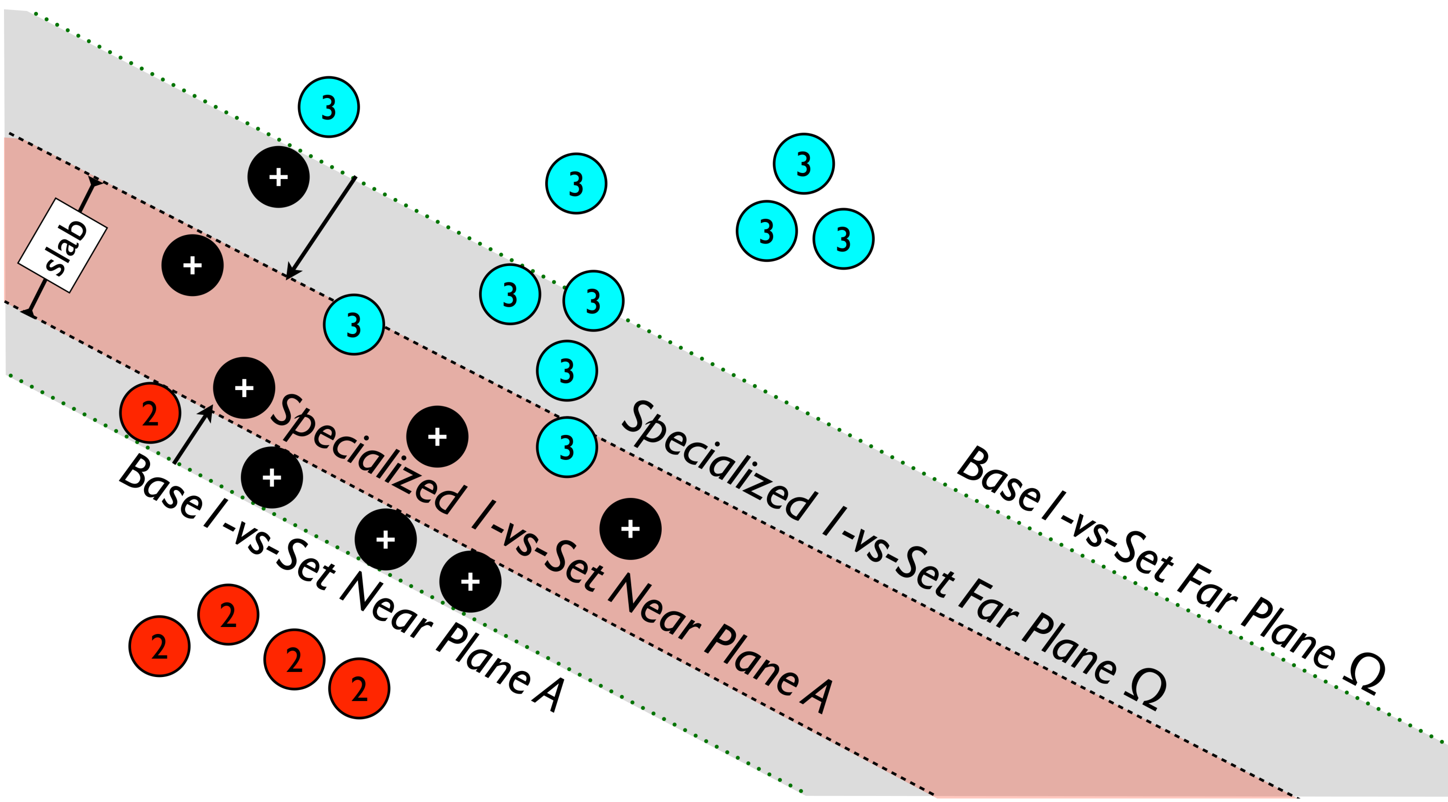
Base Linear 1-vs-Set Machine



Generalization



Specialization



Open space risk for linear slab model

δ_A Marginal distance of near plane

δ_Ω Marginal distance of far plane

δ^+ Separation needed to account for all positive data

$$\frac{\delta_\Omega - \delta_A}{\delta^+}$$

Overgeneralization risk

$$\frac{\delta^+}{\delta_\Omega - \delta_A}$$

Overspecialization risk

Open space risk for linear slab model

Two additional terms

Importance of open space around A

Importance of open space around Ω

$$R_\varsigma = \frac{\delta_\Omega - \delta_A}{\delta^+} + \frac{\delta^+}{\delta_\Omega - \delta_A} + p_A \omega_A + p_\Omega \omega_\Omega$$

Margin around A

Margin around Ω

Training and testing data

Space of positive class data: $\mathcal{P}$

Space of other known class data: $\mathcal{K}$

Positive training data: $\hat{V} = \{v_1, \ldots, v_m\}$ from $\mathcal{P}$

Negative training data: $\hat{K} = \{k_1, \ldots, k_n\}$ from $\mathcal{K}$

Unknown negatives appearing in testing: $\mathcal{U}$

Testing data: $\mathcal{T} = \{t_1, \ldots, t_z\}$, $t_i \in \mathcal{P} \cup \mathcal{K} \cup \mathcal{U}$

Sketch of the 1-vs-Set Machine training algorithm

1. Train a linear SVM f using $\hat{V}$ and $\hat{K}$

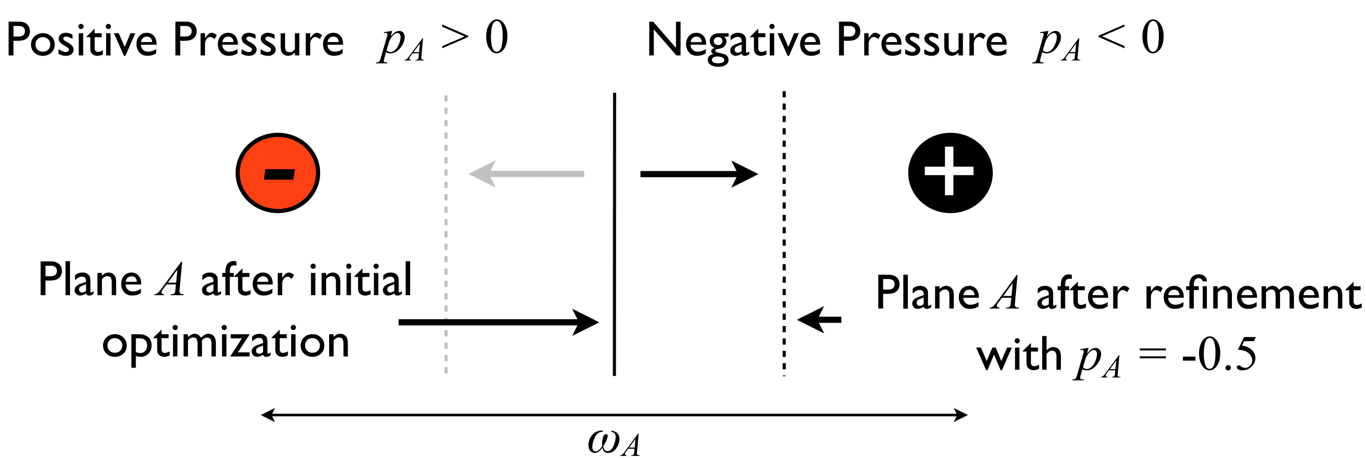
2. Generate decision scores for each training point in $\hat{V}$ and $\hat{K}$

3. Sort decision scores, where s_k is the minimum and s_j is the maximum

4. Initialize A to margin plane of f, and Ω to s_j

5. Iteratively move A to s_{k+1} or s_{k-1}, Ω to s_{j-1} or s_{j+1} to minimize $R_\varsigma(f) + \lambda_r R_\mathcal{E}$

1-vs-Set Machine Plane Refinement



1-vs-Set Machine Prediction

function PREDICT(t_x, f, A, Ω)
  if ($A \leq f(t_x)$ **and** $f(t_x) \leq \Omega$) **then Return** +1
  else Return -1
  end if
end function

What could be better about the 1-Vs-Set Machine?

- Does not inherently support multi-class open set recognition
- Does not support non-linear kernels
- Does not contain a CAP model
- **Lack of calibrated probability scores**

P_I-SVM: Modeling Probability of Inclusion

- Fit a robust single-class probability model over the positive class scores from a discriminative binary classifier
    - Binary (RBF) classifier helps discriminate the positive class from the known negative classes
    - Single-class probability model adjusts decision boundary to avoid misclassification of “unknowns”

Consider a kernelized SVM

Bias Term

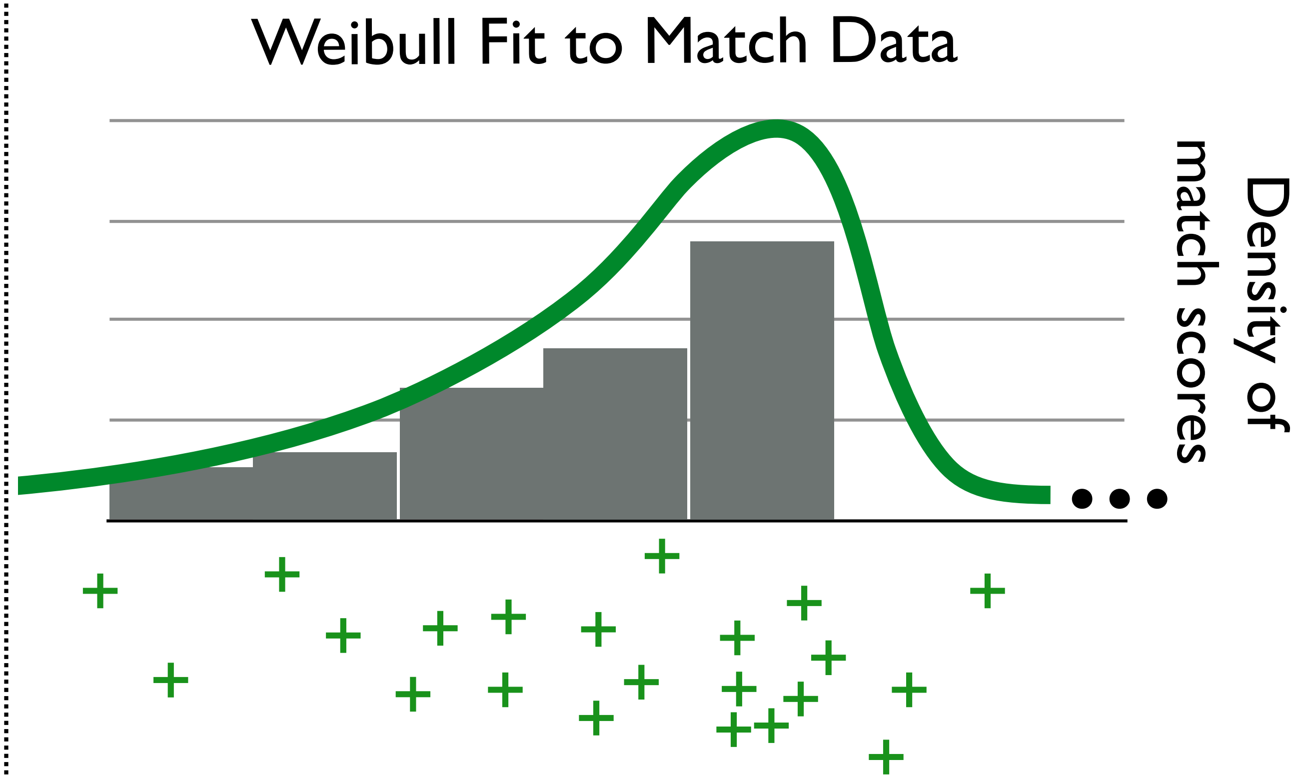
$$h(x) = \sum_{i=1}^{n} y_i \alpha_i K(x_i, x) + b$$

Support Vectors

RBF kernel

Fit model to tail of positive side of decision boundary

Non-Match Data (Negative Side)



parameters = $\theta_y = \tau_y, \kappa_y, \lambda_y$

Probability model for inclusion

$$P_I(y|x,\theta_y) = \xi\rho(y)P_I(x|y,\theta_y) = \xi\rho(y)(1 - e^{-(\frac{x-\tau_y}{\lambda_y})^{\kappa_y}})$$

Prior prob. of class y

Constant

Weibull CDF defined by θ_y

Unnormalized Posterior Estimate

If all classes and priors are known, then
Bayes' theorem yields:

$$\xi = \frac{1}{\sum_{y \in \mathcal{C}} \rho(y) P_I(x|y, \theta_y)}$$

But this isn't true for open set recognition, so we let $\xi = 1$ and treat the posterior estimate as unnormalized

Multi-class Open Set Recognition with P_I-SVM

$$y^* = \underset{y \in \mathcal{C}}{\operatorname{argmax}}\, P_I(y|x, \theta_y) \quad \textbf{subject to} \quad P_I(y^*|x, \theta_{y^*}) \geq \delta$$

Min. threshold on
class probability

Tail Size Estimation

- EVT tells us how to model extrema, but says nothing about how many samples to model
    - The difference between a tail size of 5% and a tail size of 20% can produce a difference in recognition accuracy of 15-20%
    - Need **automatic estimation**

Support Vectors as Extrema

- Support vectors are a type of extreme sampling that effectively describes the class boundary
- Is there a known parametric relationship between training data size, dimensionality, and the number of support vectors? No

Alternative: consider extrema to be the points close to the original decision boundary and count them

Tail size estimation

When > 0, some points inside the positive boundary included

Indicator Function

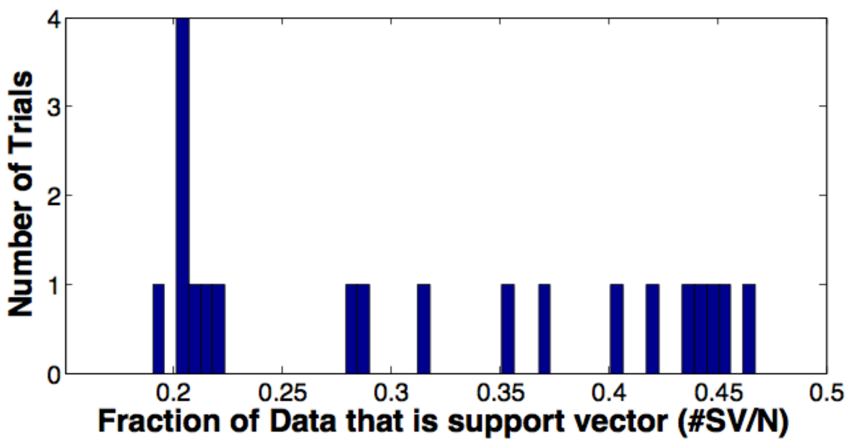
$$B^+(x;\epsilon) = \begin{cases} 1 & \text{if } h(x) \leq \epsilon \\ 0 & \text{otherwise} \end{cases} \quad \textbf{and} \quad T_\epsilon^+ = \sum_{x \in \mathcal{M}_y} B^+(x;\epsilon)$$

Positive Tail Size

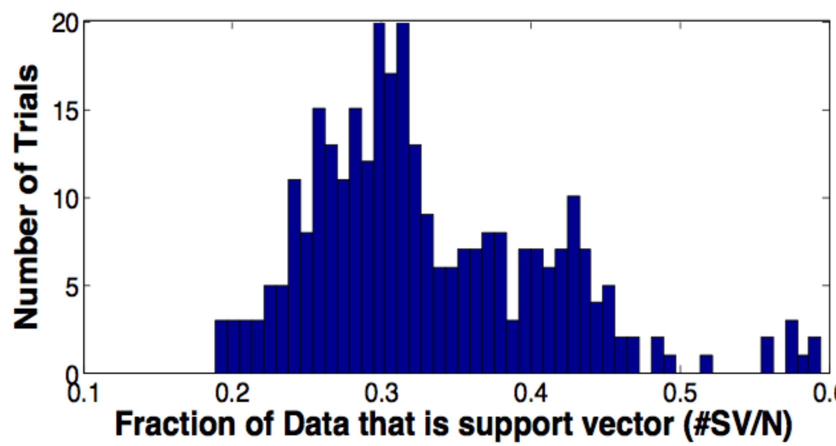
Tail size approximation: $\hat{T}_\epsilon^+ = \max(3, \psi \times |\alpha^+|)$

$\psi \in [1.25 - 2.5]$

Tail size estimation

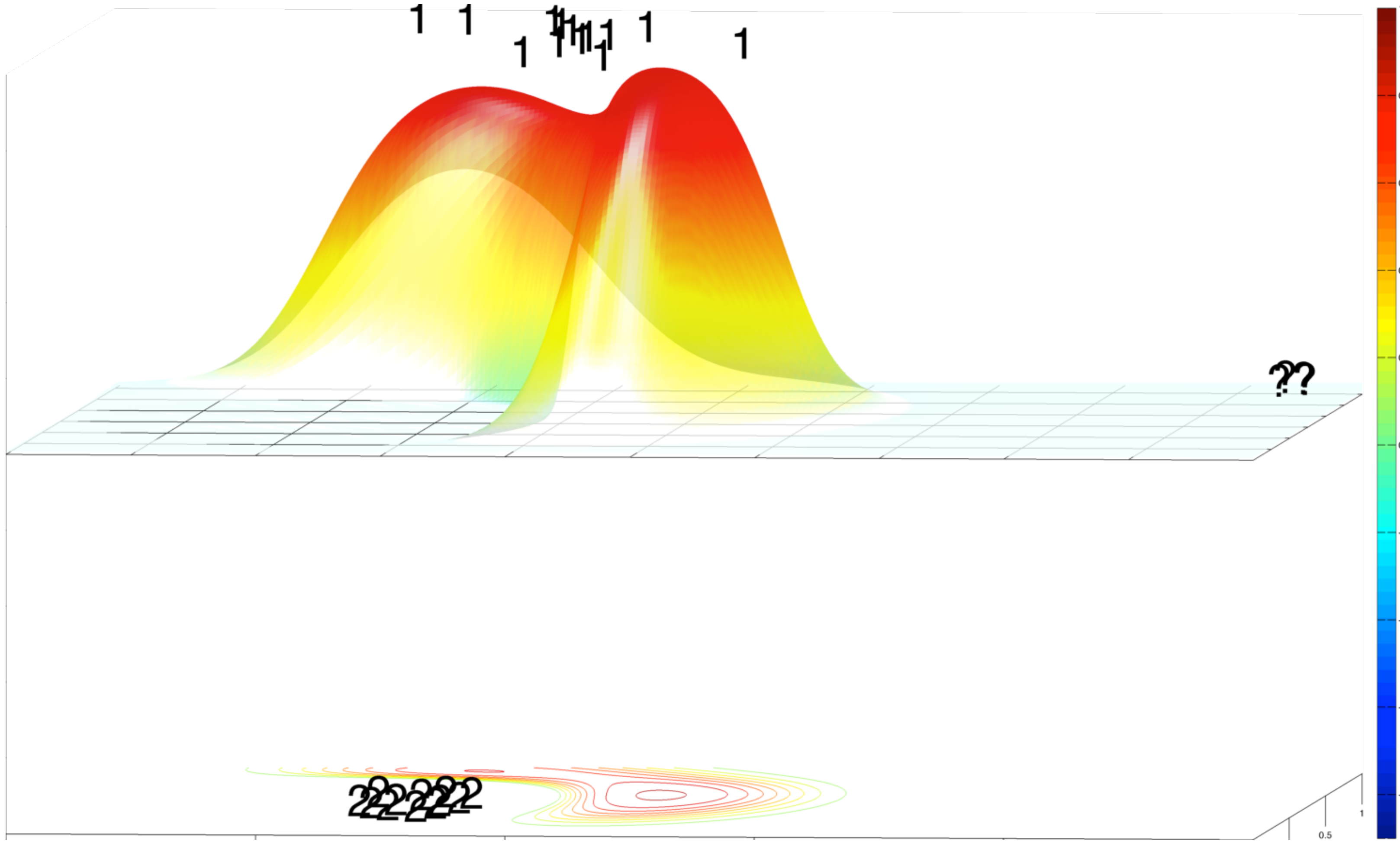


First class folds



All class folds

Normalized decision scores for P_l-SVM



P_I-SVM Implementation

Patch to LIBSVM available at:

https://github.com/ljain2/libsvm-openset

Usage: svm-train [options] training_set_file [model_file]
options:
-s svm_type : set type of SVM (default 0)
    0 -- C-SVC
    1 -- nu-SVC
    2 -- one-class SVM
    3 -- epsilon-SVR
    4 -- nu-SVR
    5 -- open-set oneclass SVM (open_set_training_file required)
    6 -- open-set pair-wise SVM (open_set_training_file required)
    7 -- open-set binary SVM (open_set_training_file required)
    8 -- one-vs-rest WSVM (open_set_training_file required)
    9 -- One-class PI-OSVM (open_set_training_file required)
    10 -- **one-vs-all PI-SVM (open_set_training_file required)**

Is PI-SVM what we're looking for for open set recognition?

- Pros:

  + Supports multi-class open set recognition

  + Better generalization than the 1-vs-Set Machine

- Cons:

  - One-sided calibration model (just probability of inclusion)

  - Does not make use of a CAP model

NN+CAP

Let d_x be the distance to the nearest neighbor of x

Let $d_x > \tau \Rightarrow p_a(x) = 0$ and $p_a(x) = \frac{|\tau - d_x|}{\tau}$

In a multi-class setting, this results in a thresholded NN algorithm that can reject an input as unknown.

NN+CAP

- Pros:

  + With sufficiently dense sampling, NN+CAP reduces to NN

  + Limiting error of no more than twice the Bayes error rate

  + Simple to train

- Cons:

  - **Weak probability model**

The Weibull-calibrated SVM (W-SVM)

- Binary SVMs are better than 1-Class SVMs - how do they fit into the context of CAP models?
- Unfortunately, the decision score isn't a canonical sum. But calibration is possible (Hoffman et al. Annals of Stat. 2008):

1. Collect all positive coefficients in one sum
2. Collect all negative coefficients into another sum
3. Split the bias between them
4. View SVM as applying a decision rule over which is more similar

T. Hofmann, B. Schölkopf, and A. J. Smola. Kernel methods in machine learning. Annals of Stat., 2008.

Binary RBF SVM incorporating a CAP model

- Combine probabilities computed for both 1-class and binary RBF SVMs
- 1-class SVM CAP model is a conditioner

could be very small

if $P_O(y|x) > \delta_\tau$, then

consider $P_O(y|x)$

else

reject

Dual tail fitting

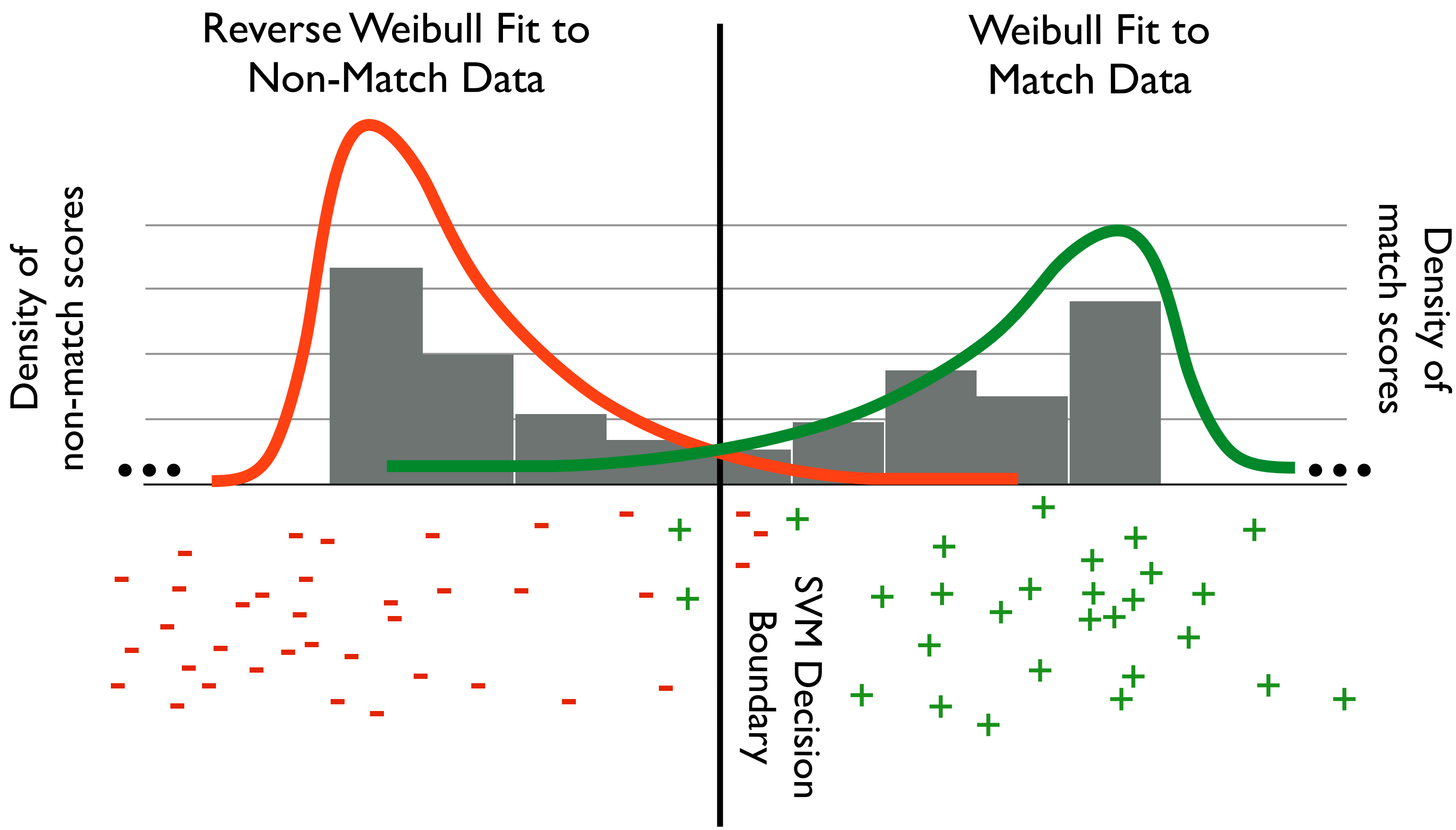
Separating positive and negative data is useful

Assume a set of known classes $\mathcal{Y}$

For a class $y \in Y$, we can use positive scores from y to estimate $P^{+}(y|x)$.

We can use negative scores from other known classes to estimate $P^{-}(\mathcal{Y} \setminus y \mid x)$.

Dual tail fitting



Dual tail fitting

Closed set scenario: $P^{+}(y|x) = 1 - P^{-}(\mathcal{Y} \setminus y \mid x)$

In an open set scenario, we can't make the above assumption.

To minimize open set risk, P^{+} and P^{-} are considered only when $P_{O}(y|x) > \delta_{\tau}$

EVT Parameters

- Reverse Weibull and Weibull are defined by three parameters
  - location ν, scale λ, and shape κ
- Maximum Likelihood Estimation to estimate the best fits for η and ψ
  - ν_η, λ_η, κ_η
  - ν_ψ, λ_ψ, κ_ψ

Two independent estimates for $P(\, y \mid f(x))$

Weibull CDF from match data

$$P_\eta(y|f(x)) = 1 - e^{-(\frac{f(x)-\nu_\eta}{\lambda_\eta})^{\kappa_\eta}}$$

Reverse Weibull CDF from non-match data

$$P_\psi(y|f(x)) = e^{-(\frac{f(x)-\nu_\psi}{\lambda_\psi})^{\kappa_\psi}}$$

Combining probability estimates

Two options:

$P\eta \times P\psi$: the probability that the input is from the positive class AND NOT from any of the known negative classes.

$P\eta + P\psi$: either a positive OR NOT a known negative.

For open set recognition, $P\psi$ should be modulated by other supporting evidence of the sample being positive. Product is the preferred combo.

Multi-class W-SVM recognition

free parameter

Indicator variable: $\iota_y = 1$ if $P_O(y|x) > \delta_\tau$

$$y^* = \underset{y \in \mathcal{Y}}{\operatorname{argmax}}\, P_{\eta,y}(x) \times P_{\psi,y}(x) \times \iota_y$$

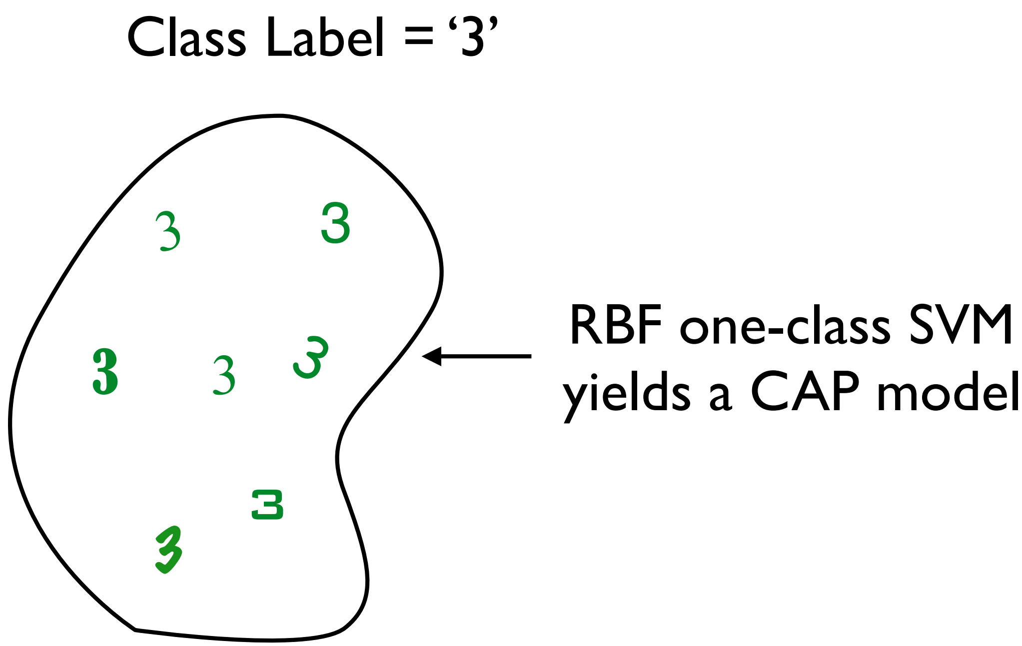
$$\text{subject to } P_{\eta,y^*}(x) \times P_{\psi,y^*}(x) \geq \delta_R$$

free parameter

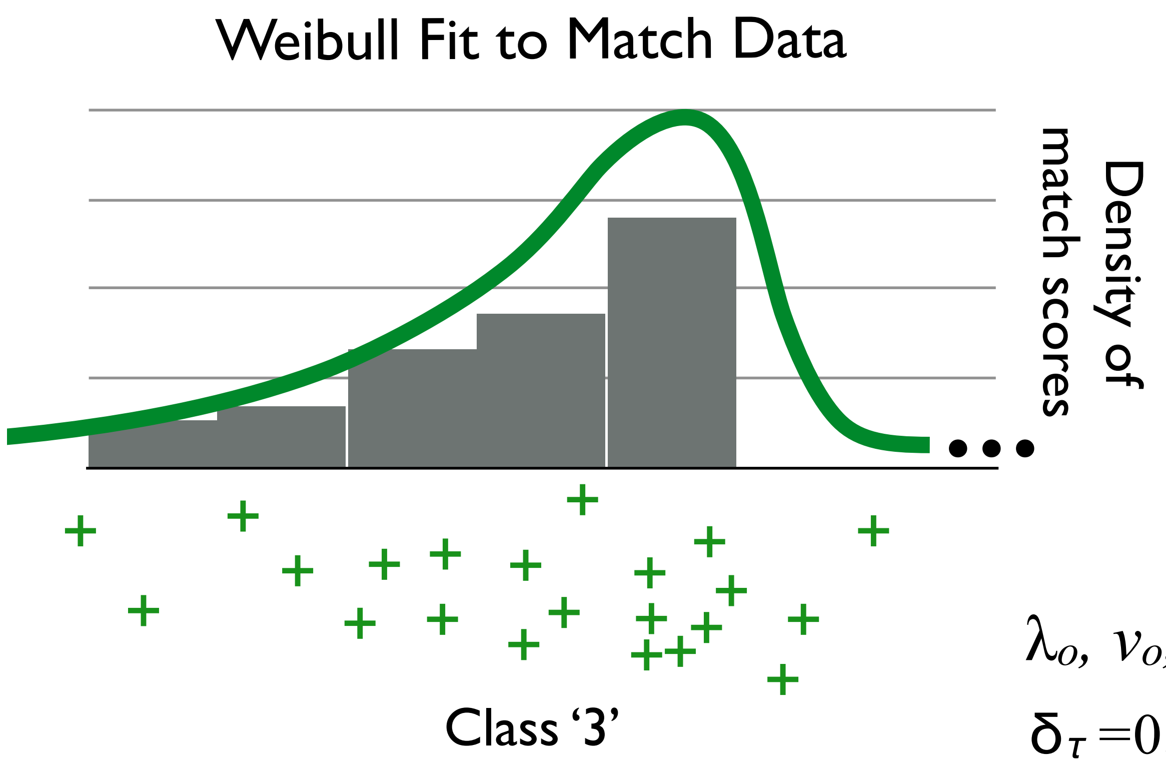
Training a W-SVM Step-by-Step

- For simplicity, let's focus on a single class ("3")
- Two SVM models (1-class and binary)
- Three EVT distribution fits
- The collection of SVM models, EVT distribution parameters, and thresholds constitute the W-SVM.

Step 1: Train a 1-class SVM f^o



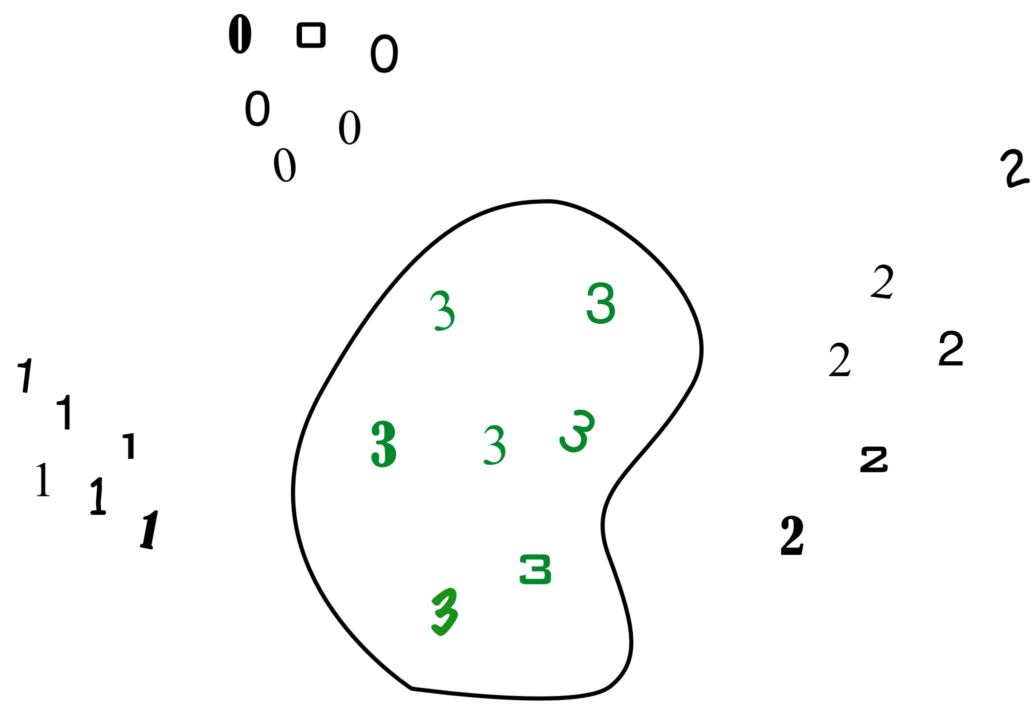
Step 2: Fit Weibull over tail of scores from f^o



$\lambda_o,\ \nu_o,\ \kappa_o$

$\delta_\tau = 0.001$

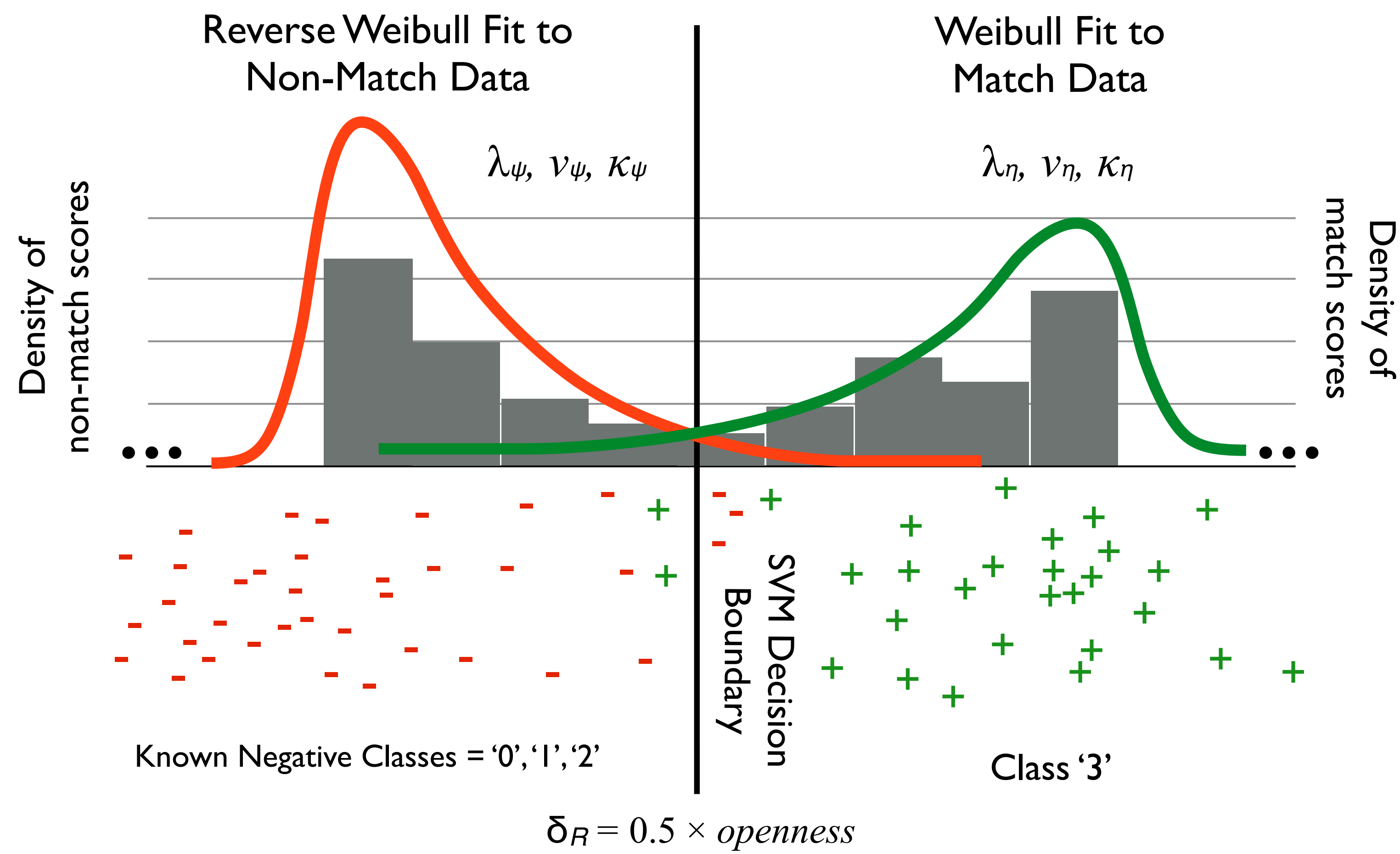
Step 3: Train a binary SVM f



Class Label = '3'

Known Negative Classes = '0', '1', '2'

Step 4: Fit EVT distributions over tails of scores from f



W-SVM testing (known class)

- Let’s focus on the class we just trained for (“3”)
- Six steps are necessary to test the input
- Assume four known classes (“0”, “1”, “2”, “3”)

Step 1: Apply 1-class SVM CAP model for all known classes

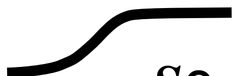
Input: x = 3

$f^{o}_{0}(\mathbf{x}) = s_0 \quad f^{o}_{1}(\mathbf{x}) = s_1$

$f^{o}_{2}(\mathbf{x}) = s_2 \quad f^{o}_{3}(\mathbf{x}) = s_3$

Step 2: Normalize all 1-class SVM scores using EVT models

$\lambda_{o,0},\ \nu_{o,0},\ \kappa_{o,0}$



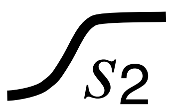
s_0

$\lambda_{o,1},\ \nu_{o,1},\ \kappa_{o,1}$

s_1

Apply CDF for each class to each score $\longrightarrow$ Probability model for test instance: P_o

$\lambda_{o,2},\ \nu_{o,2},\ \kappa_{o,2}$



s_2

$\lambda_{o,3},\ \nu_{o,3},\ \kappa_{o,3}$

s_3

Step 3: Test probabilities

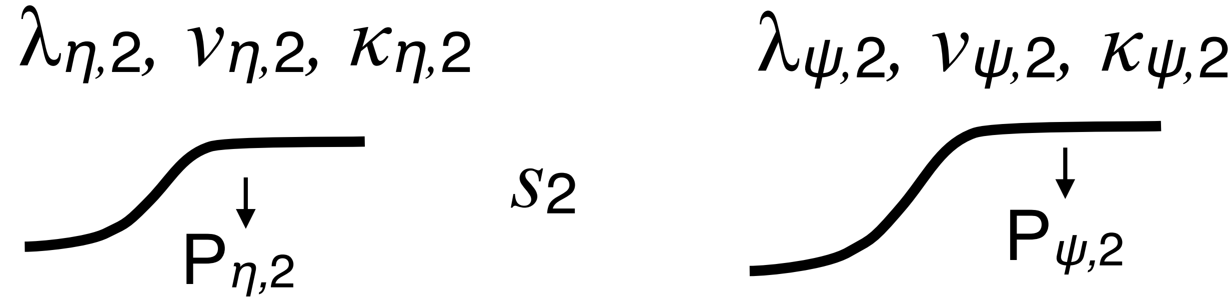
$P_o(0|x) < \delta_\tau$, $l_0 = 0$; $P_o(1|x) < \delta_\tau$, $l_1 = 0$;

$P_o(2|x) > \delta_\tau$, $l_2 = 1$; $P_o(3|x) > \delta_\tau$, $l_3 = 1$

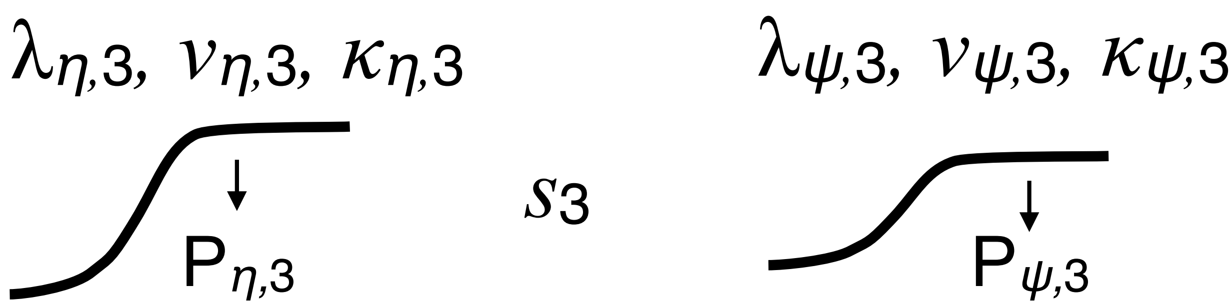
Step 4: Apply binary SVMs

$$f_2(\mathbf{x}) = s_2 \qquad f_3(\mathbf{x}) = s_3$$

Step 5: Normalize all binary SVM scores using EVT match and non-match models



Apply 2 CDFs per class for each score



Step 6: Fuse and test probabilities

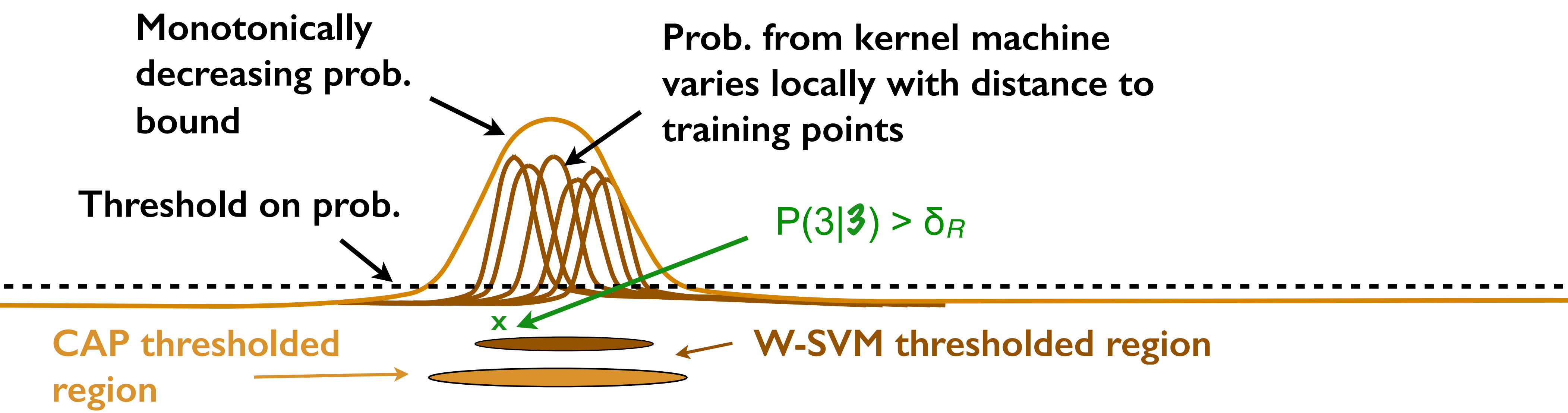
$P_{\eta,0}(x) \times P_{\psi,0}(x) \times \iota_0 = 0 < \delta_R$

$P_{\eta,1}(x) \times P_{\psi,1}(x) \times \iota_1 = 0 < \delta_R$

$P_{\eta,2}(x) \times P_{\psi,2}(x) \times \iota_2 = 0.001 < \delta_R$

$P_{\eta,3}(x) \times P_{\psi,3}(x) \times \iota_3 = 0.877 > \delta_R$

Models for class ‘3’ and the data point for this example



W-SVM testing (unknown class)

- Assume four known classes (“0”, “1”, “2”, “3”)
- Consider as input a member of a class that is different from the training data (“Q”)
  - This point will fall outside of the CAP thresholded region (*i.e.*, it exists in open space)
- Four steps are necessary to reject the input

Step 1. Apply 1-class SVM CAP model for all known classes

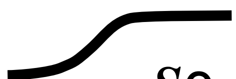
Input: x = **Q**

$f^{o}_{0}(\mathsf{x}) = s_0 \quad f^{o}_{1}(\mathsf{x}) = s_1$

$f^{o}_{2}(\mathsf{x}) = s_2 \quad f^{o}_{3}(\mathsf{x}) = s_3$

Step 2. Normalize all 1-class SVM scores using EVT models

$\lambda_{o,0},\ \nu_{o,0},\ \kappa_{o,0}$



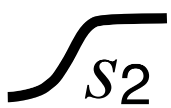
s_0

$\lambda_{o,1},\ \nu_{o,1},\ \kappa_{o,1}$

s_1

Apply CDF for each class to each score $\longrightarrow$ Probability model for test instance: P_o

$\lambda_{o,2},\ \nu_{o,2},\ \kappa_{o,2}$



s_2

$\lambda_{o,3},\ \nu_{o,3},\ \kappa_{o,3}$

s_3

Step 3: Test probabilities

$P_o(0|x) < \delta_\tau,\ l_0 = 0$; $P_o(1|x) < \delta_\tau,\ l_1 = 0$;

$P_o(2|x) < \delta_\tau,\ l_2 = 0$; $P_o(3|x) < \delta_\tau,\ l_3 = 0$

Step 4: Apply indicator variables to binary SVMs

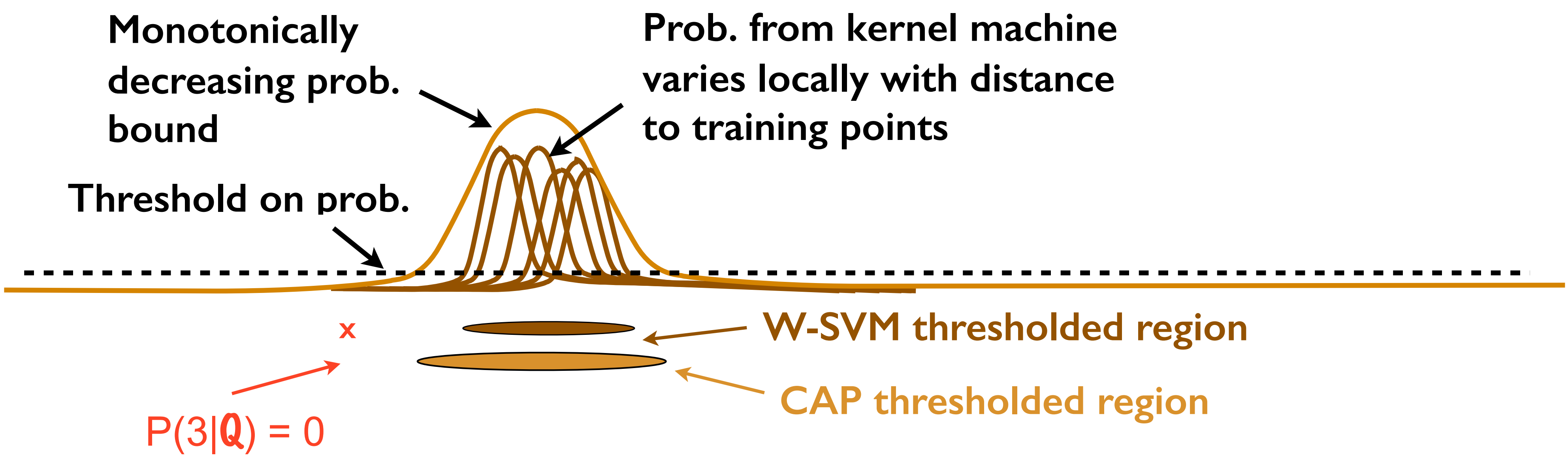
$$P_{\eta,0}(x) \times P_{\psi,0}(x) \times \iota_0 = 0 < \delta_R$$

$$P_{\eta,1}(x) \times P_{\psi,1}(x) \times \iota_1 = 0 < \delta_R$$

$$P_{\eta,2}(x) \times P_{\psi,2}(x) \times \iota_2 = 0 < \delta_R$$

$$P_{\eta,3}(x) \times P_{\psi,3}(x) \times \iota_3 = 0 < \delta_R$$

Models for class '3' and the data point for this example



W-SVM Implementation

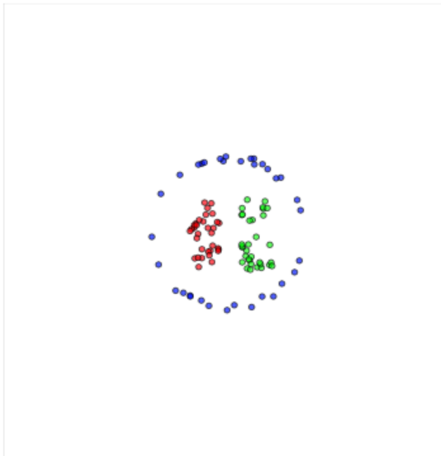
Patch to LIBSVM available at:

https://github.com/ljain2/libsvm-openset

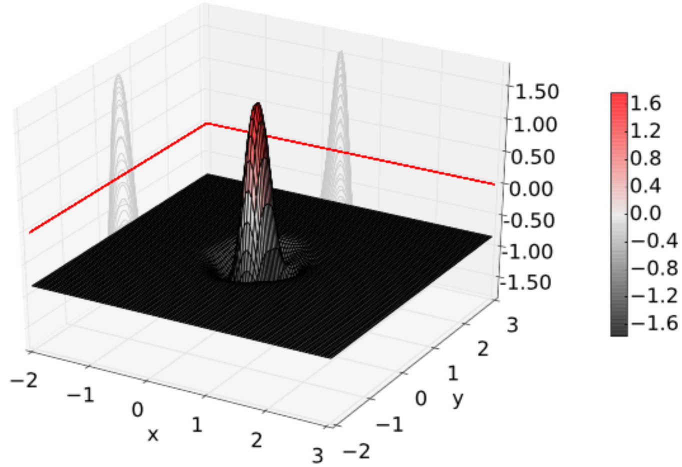
Usage: svm-train [options] training_set_file [model_file]
options:
-s svm_type : set type of SVM (default 0)
- 0 -- C-SVC
- 1 -- nu-SVC
- 2 -- one-class SVM
- 3 -- epsilon-SVR
- 4 -- nu-SVR
- 5 -- open-set oneclass SVM (open_set_training_file required)
- 6 -- open-set pair-wise SVM (open_set_training_file required)
- 7 -- open-set binary SVM (open_set_training_file required)
- 8 -- **one-vs-rest WSVM (open_set_training_file required)**
- 9 -- One-class PI-OSVM (open_set_training_file required)
- 10 -- one-vs-all PI-SVM (open_set_training_file required)

Specialized Support Vector Machine (SSVM)

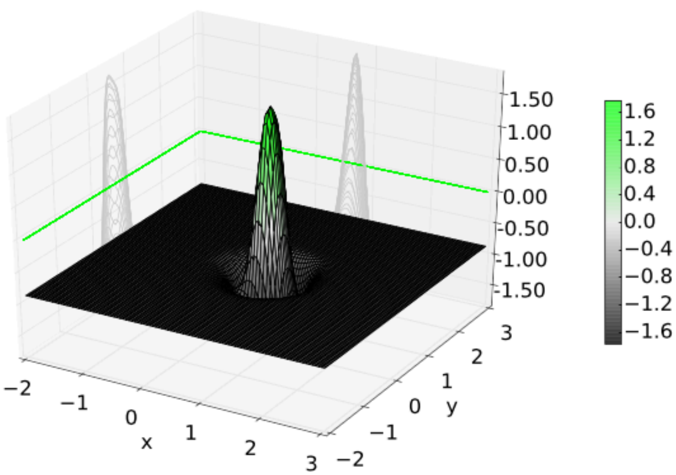
Junior, Wainer and Rocha, arXiv 2016



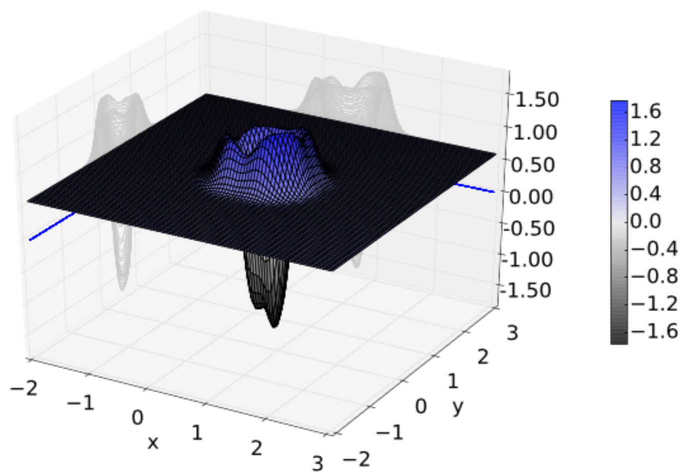
Boat dataset with 3 classes: red (the central class to the left), green (the central class to the right), and blue (the class with the ring shape).



(a) Class 1 (red). $b = -0.832$.



(b) Class 2 (green). $b = -0.86$.



(c) Class 3 (blue). $b = +0.594$.

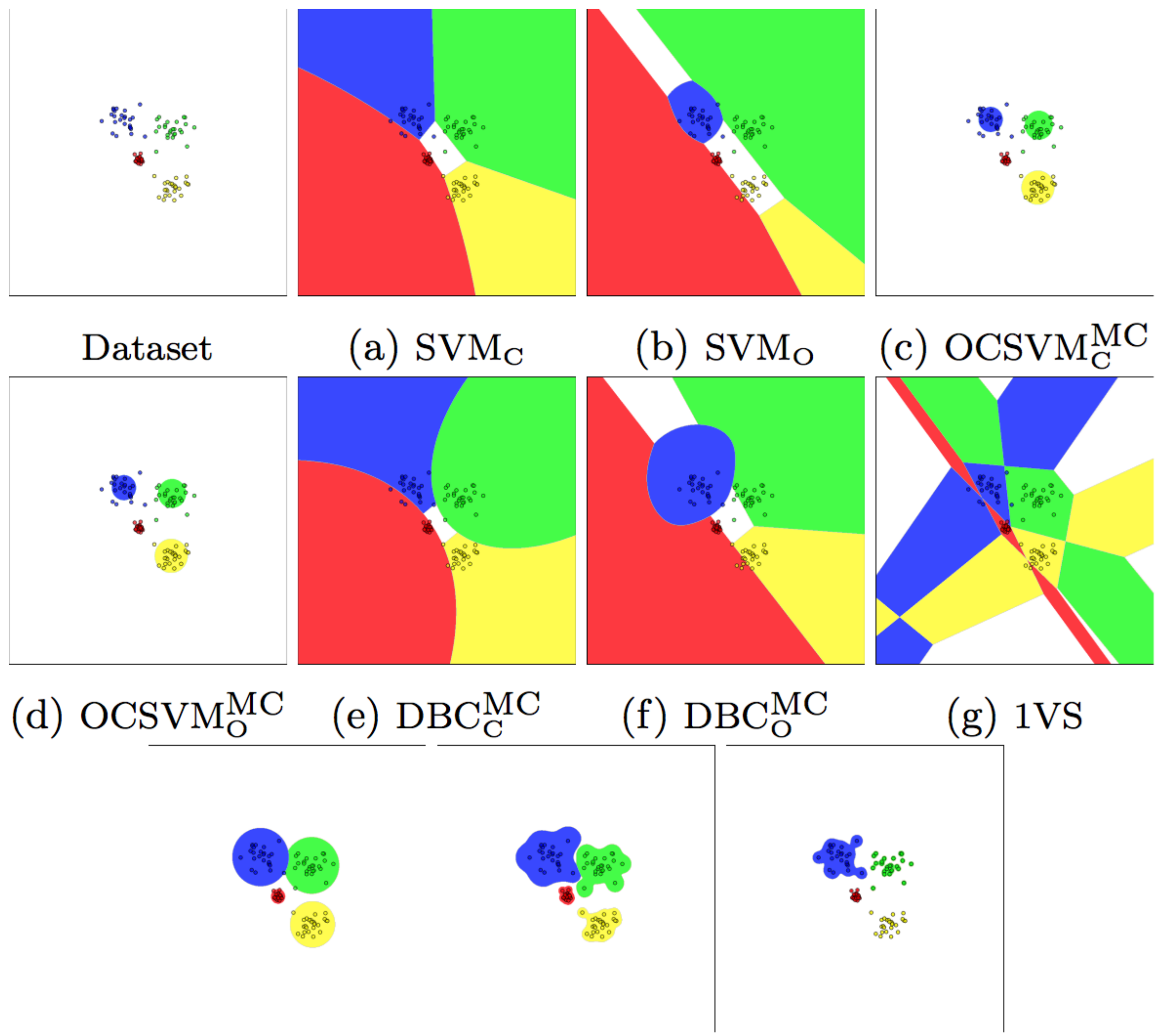
Specialized Support Vector Machine (SSVM)

Ensure bounded positively labeled open space by using an RBF kernel and **forcing the bias to be negative**

$$b' \in \left\{ -\frac{|b|(2^i - 1)}{2^{|b|} - 1}, i \in (0, |b|] \right\},$$

Determined via open set grid search procedure

Specialized Support Vector Machine (SSVM)



Dataset (a) SVM_C (b) SVM_O (c) $OCSVM_C^{MC}$

(d) $OCSVM_O^{MC}$ (e) DBC_C^{MC} (f) DBC_O^{MC} (g) 1VS

(h) WSVM (i) $SSVM_C$ (j) $SSVM_O$

How can we evaluate open set recognition in a controlled manner?

Accuracy as a statistic for open set problems

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Imagine the following case:

1/100 TP correct
100,000/100,000 TN correct
99.9% accuracy!

F-measure as a statistic for open set problems

Consistent point of comparison across inconsistent precision and recall numbers:

$$\text{F-measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Open Set Object Recognition

Cross-data set methodology*

Training: Caltech 256



known classes

Testing: Caltech 256 + ImageNet



known classes +



unknown classes

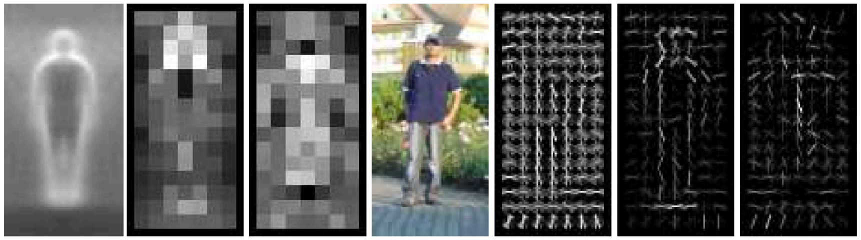
Open Universe of 88 classes: 1 positive class, n training classes, 87 negative testing classes (532,400 images)

Open Universe of 212 classes: 1 positive class, n training classes, 211 negative testing classes (13,610,400 images)

A. Torralba and A. A. Efros, “Unbiased Look at Dataset Bias,” in IEEE CVPR 2011.

Features

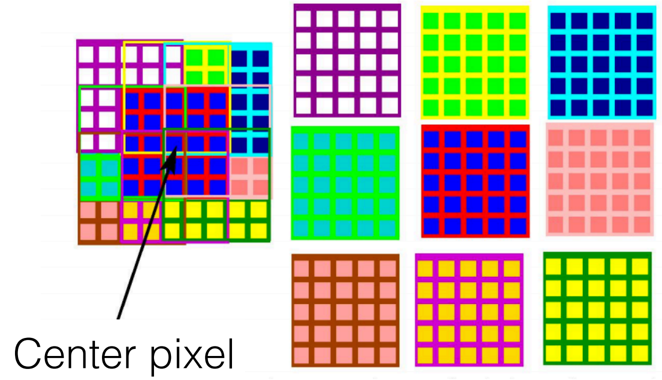
Histogram of Oriented Gradients



(Dalal and Triggs 2005) © 2005 IEEE

N. Dalal and B. Triggs, “Histograms of Oriented Gradients for Human Detection,” in IEEE CVPR, 2005

LBP-like descriptor



A. Sapkota, B. Parks, W.J. Scheirer, and T. Boult, “FACE-GRAB: Face Recognition with General Region Assigned to Binary Operator

1-vs-Set Machine vs. Typical SVMs

2-tailed paired t-test	binary 1-vs-Set	binary linear	binary RBF	1-class 1-vs-Set	1-class linear	1-class RBF
binary 1-vs-Set (HOG 88)		**	**	**	**	**
binary linear (HOG 88)	—		—	++	++	++
binary RBF (HOG 88)	—	++		++	++	++
1-class 1-vs-Set (HOG 88)	—	—	—		**	—
1-class linear (HOG 88)	—	—	—	—		—
1-class RBF (HOG 88)	—	—	—	—	++	
binary 1-vs-Set (HOG 212)		**	*	**	**	**
1-class 1-vs-Set (HOG 212)	—	—	—		—	*
binary 1-vs-Set (LBP-like 88)		**	**	**	**	**
1-class 1-vs-Set (LBP-like 88)	—	—	—		**	—
binary 1-vs-Set (LBP-like 212)		*	—	**	**	**
1-class 1-vs-Set (LBP-like 212)	—	—	—		**	—

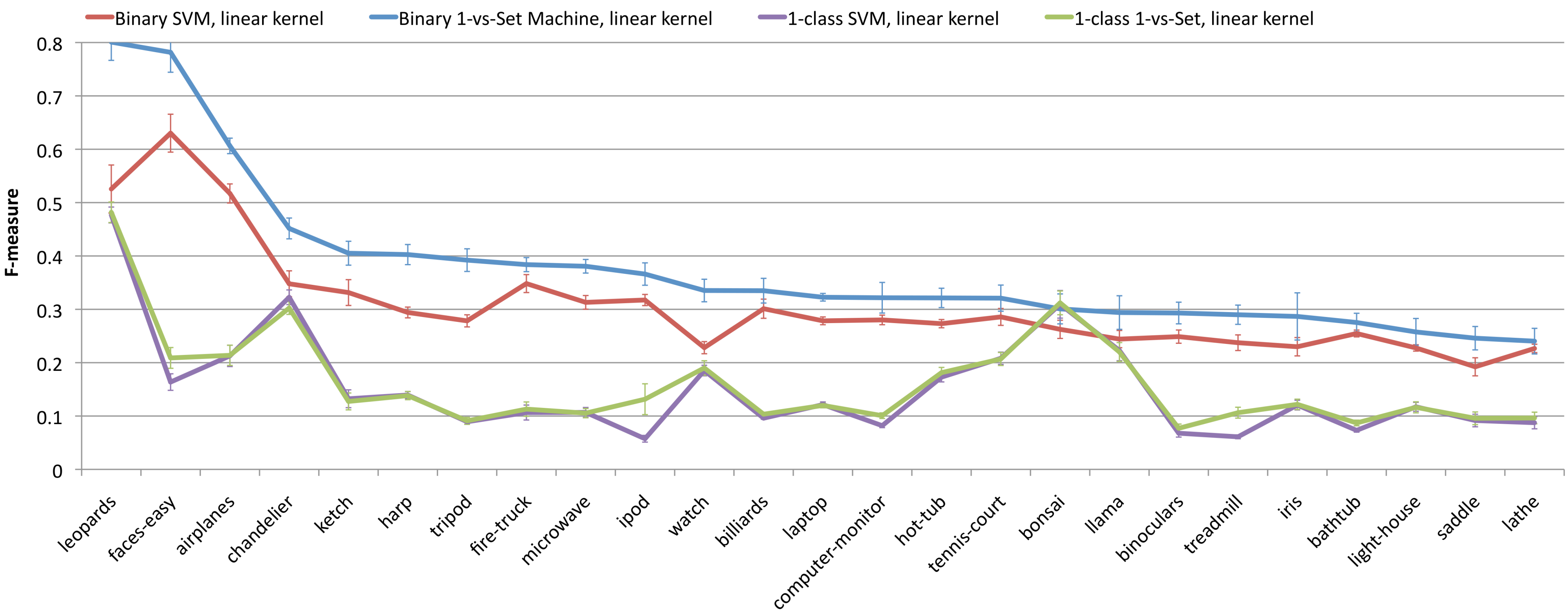
** 1-vs-Set Machine is statistically significant at $p < 0.01$

* 1-vs-Set Machine is statistically significant at $p < 0.05$

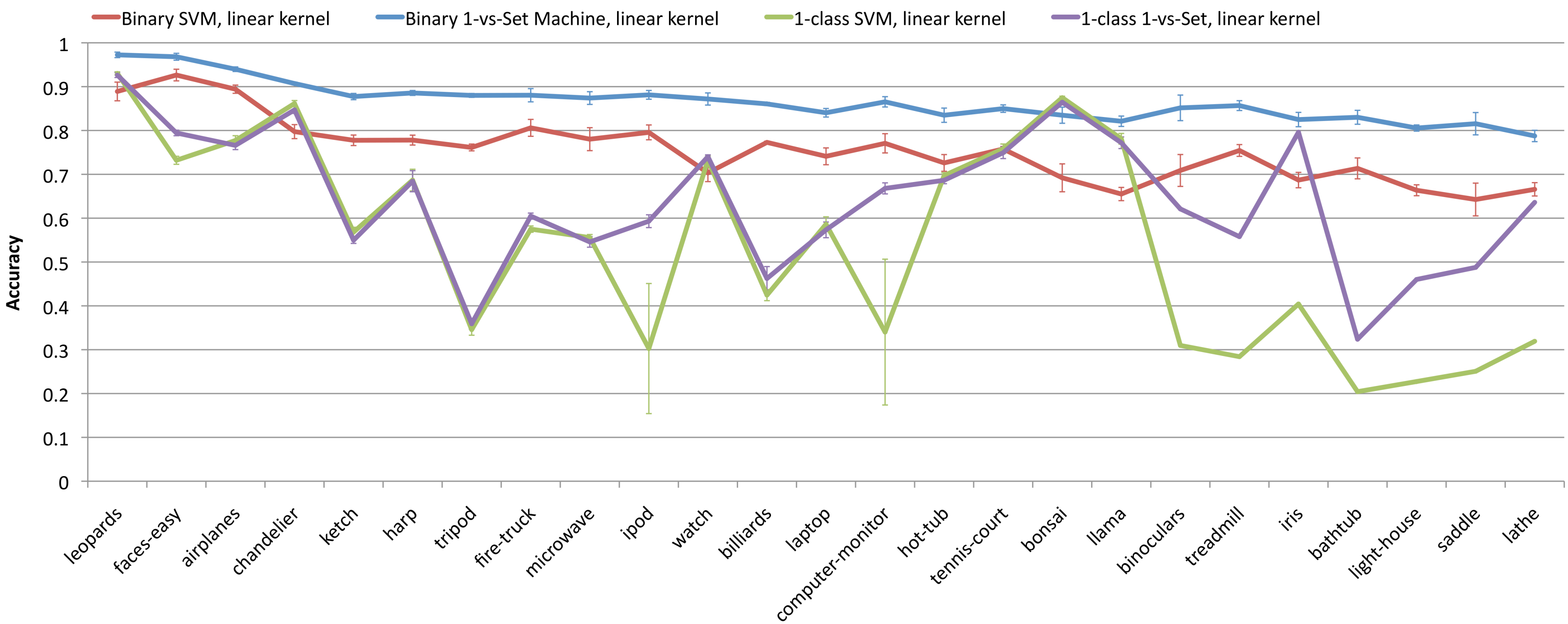
++ Baseline Machine is statistically significant at $p < 0.01$

— No statistical significance

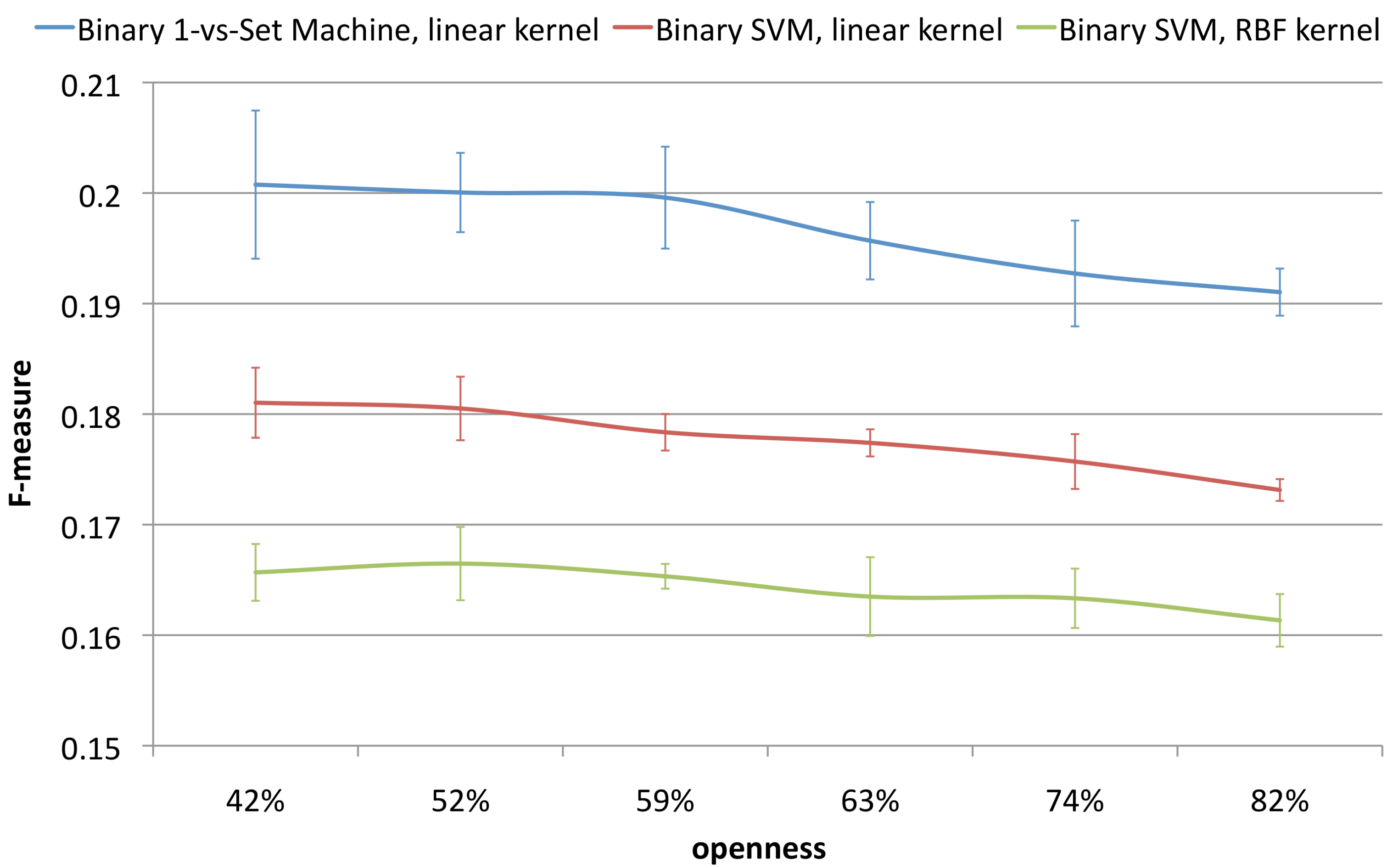
Top 25 classes for the open universe of 88 classes



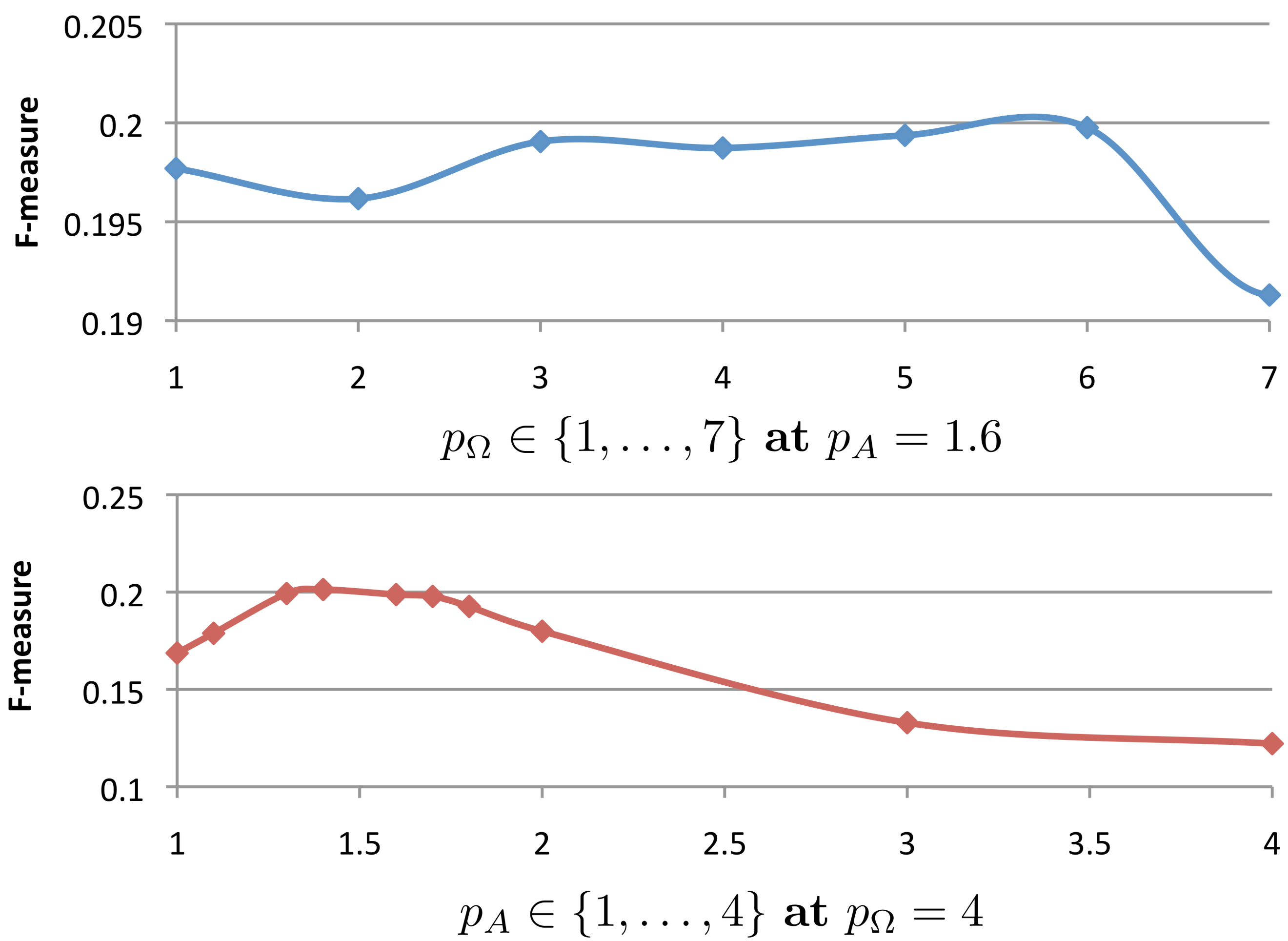
Top 25 classes for the open universe of 88 classes



F-measure as a function of openness



Near and far plane pressures for open universe of 88 classes

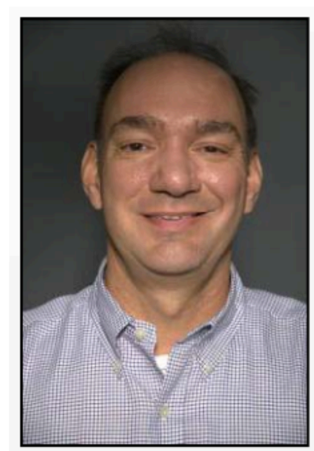


The second plane has an impact on recognition performance

Biometric Verification

Does this incoming sample match the one in our system?

New Sample



Stored Image

Answer: Verified or Not Verified

P. J. Phillips et al., Multiple Biometrics Grand Challenge

Score Distributions

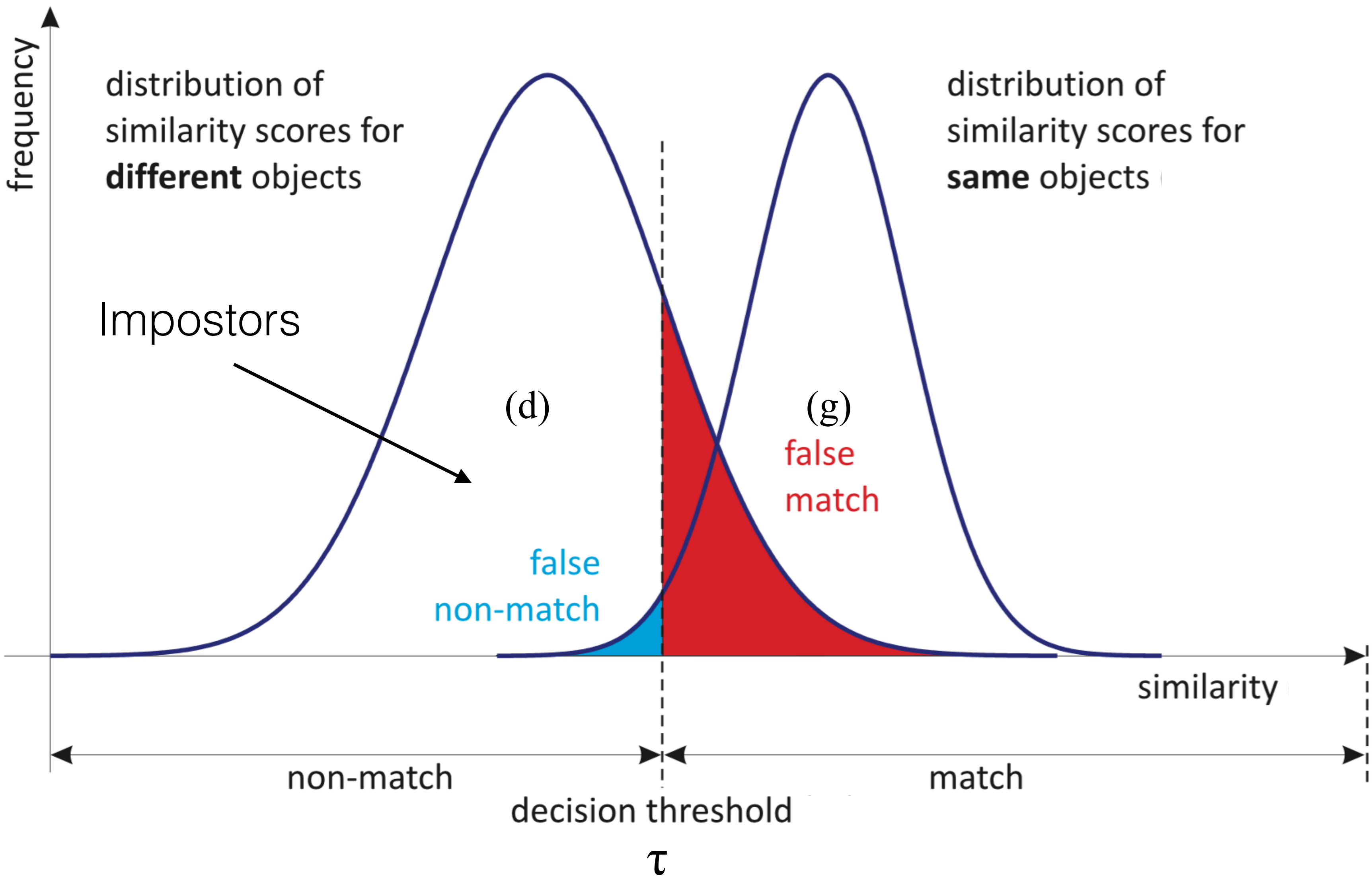


Image Credit: A. Czajka

Open Set Face Verification

Labeled Faces in the Wild

Genuine Pair

Impostor Pair

Impostor Pair

Impostor Pair

Gallery classes: 12 people with at least 50 images

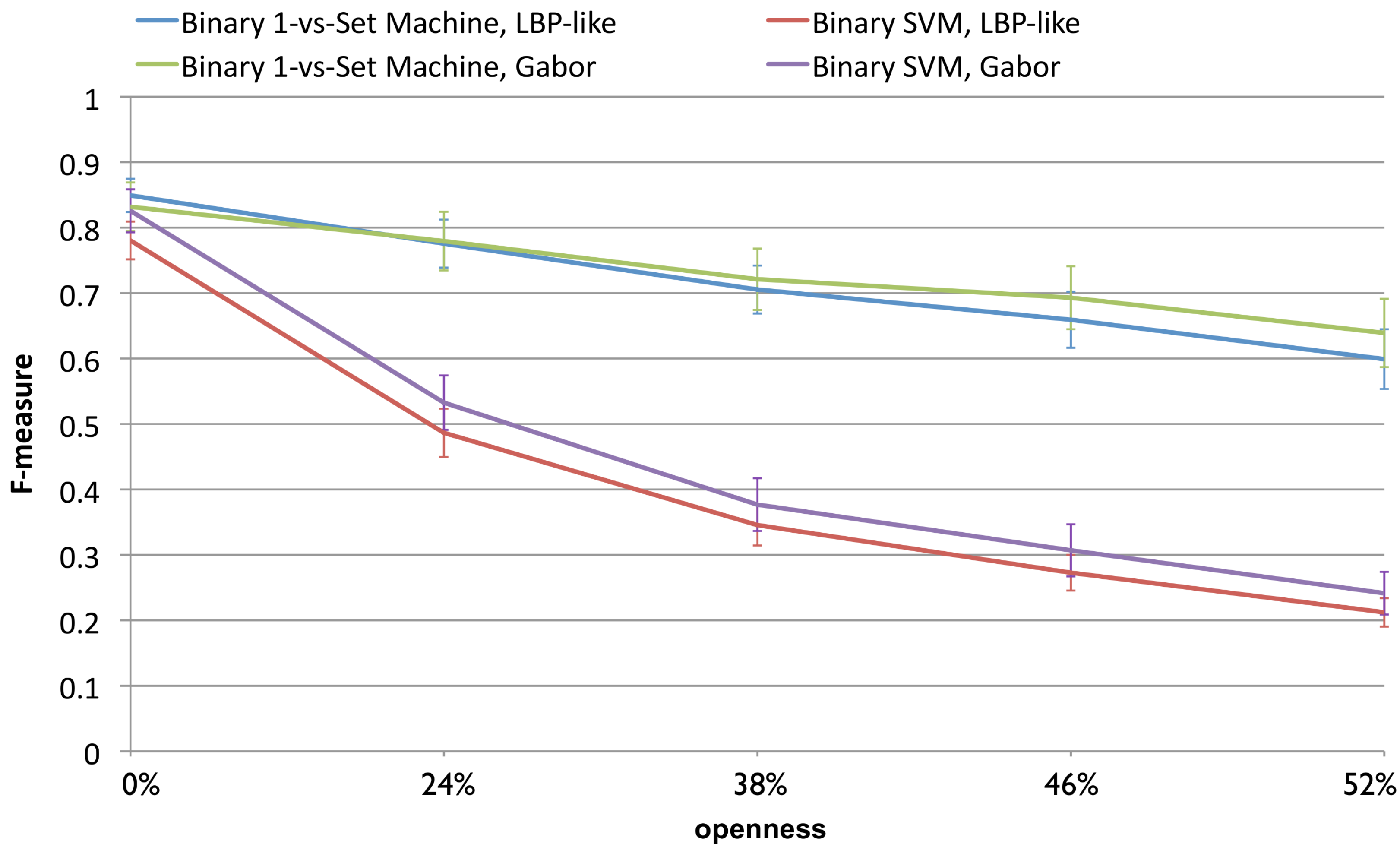
Impostor classes: 82 other people in LFW

1,316 test images across all classes

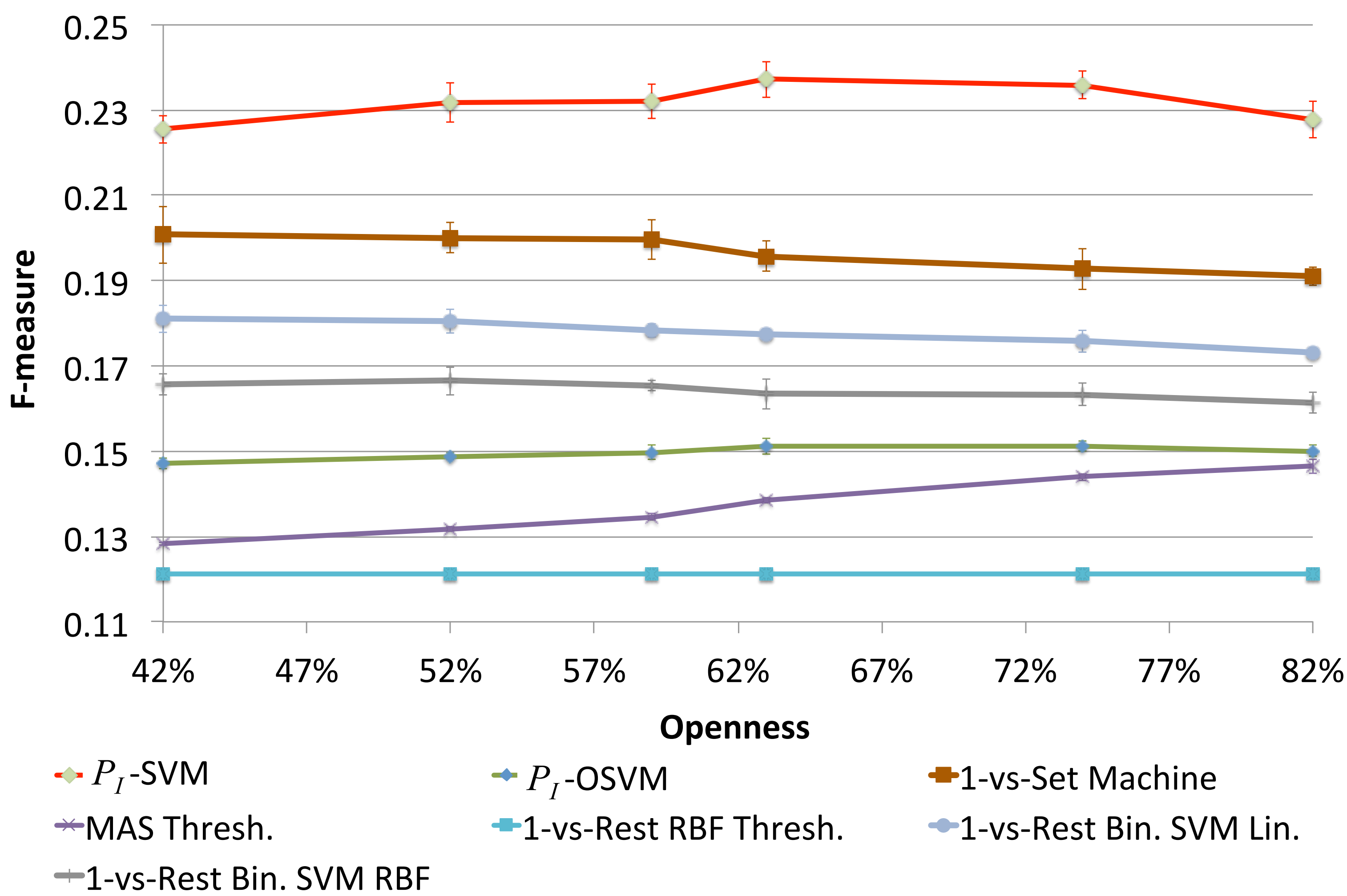
Features: LBP-like and Gabor*

N. Pinto, J. J. DiCarlo, and D. D. Cox, “How Far Can You Get with a Modern Face Recognition Test Set Using Only Simple Features?” in IEEE CVPR, 2009.

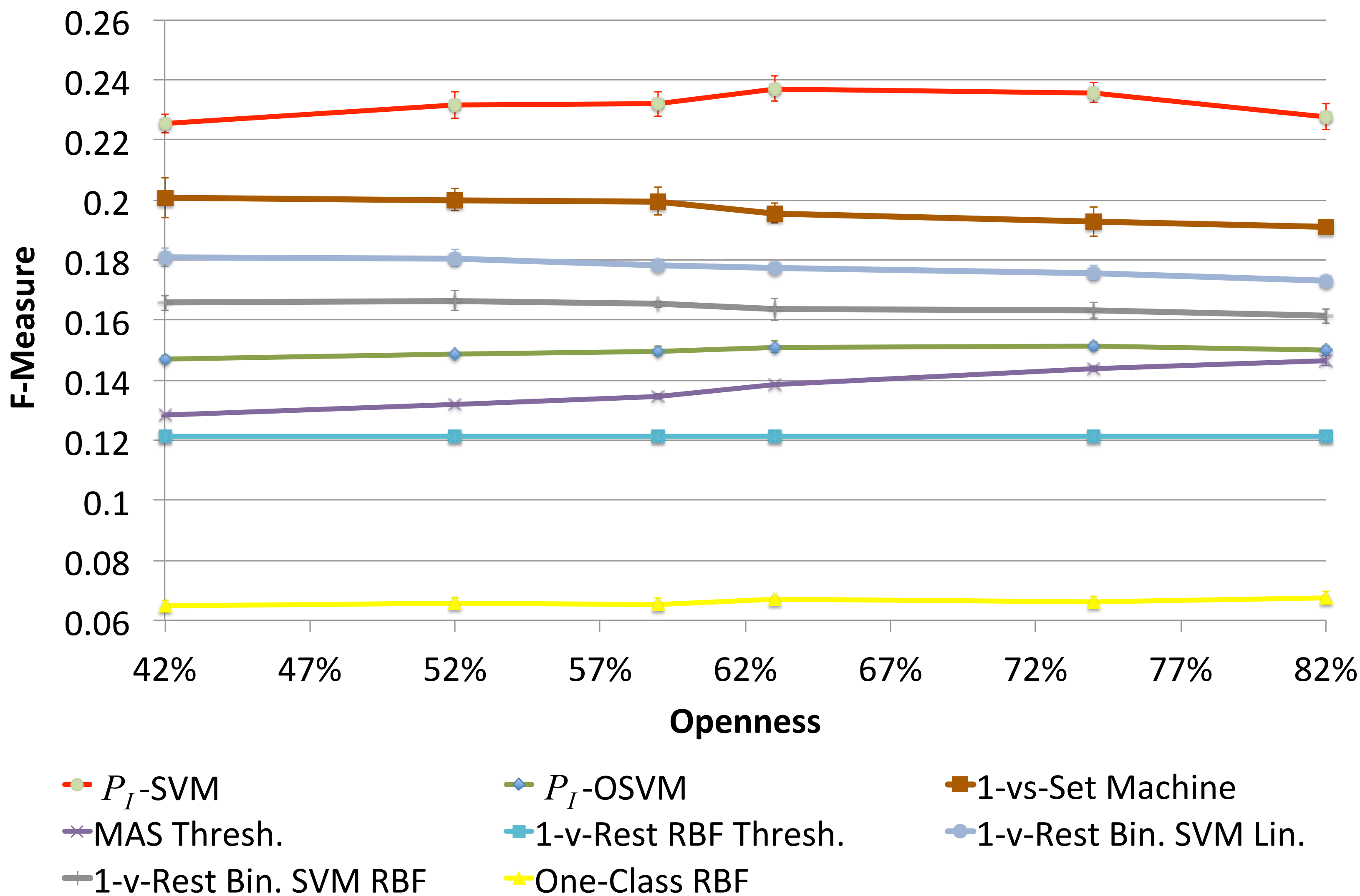
Open set face verification



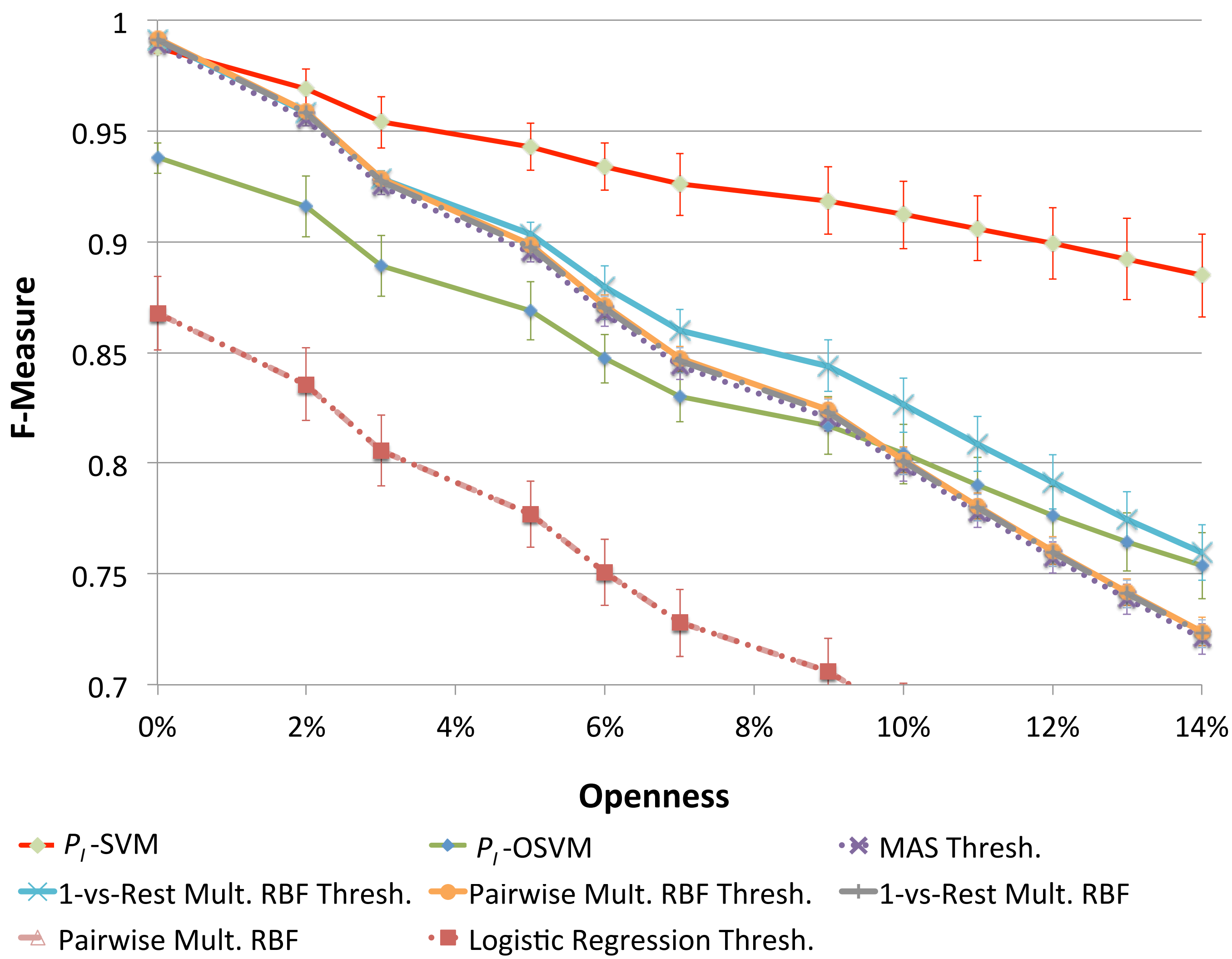
P_I-SVM Object Recognition



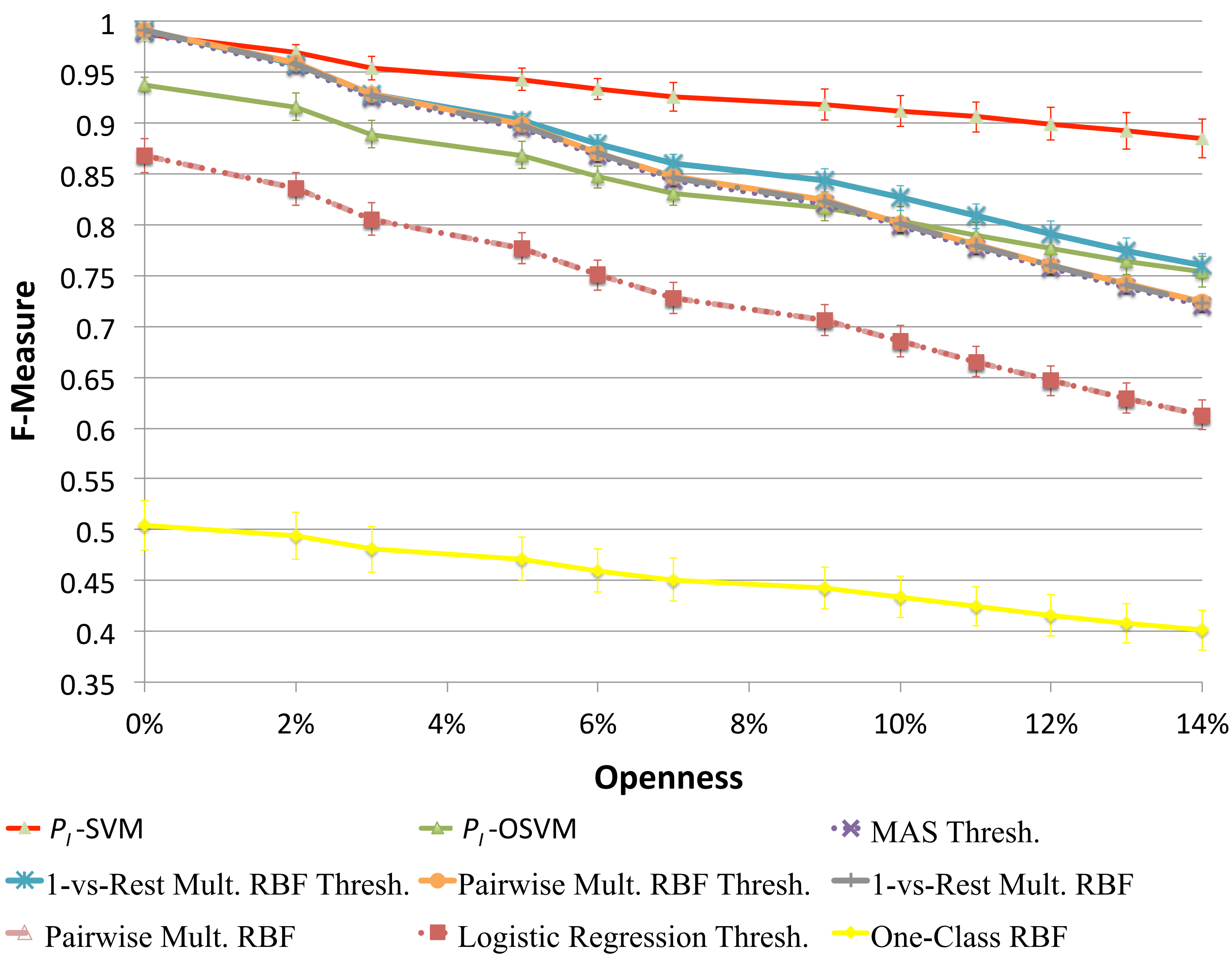
P_I-SVM Object Recognition



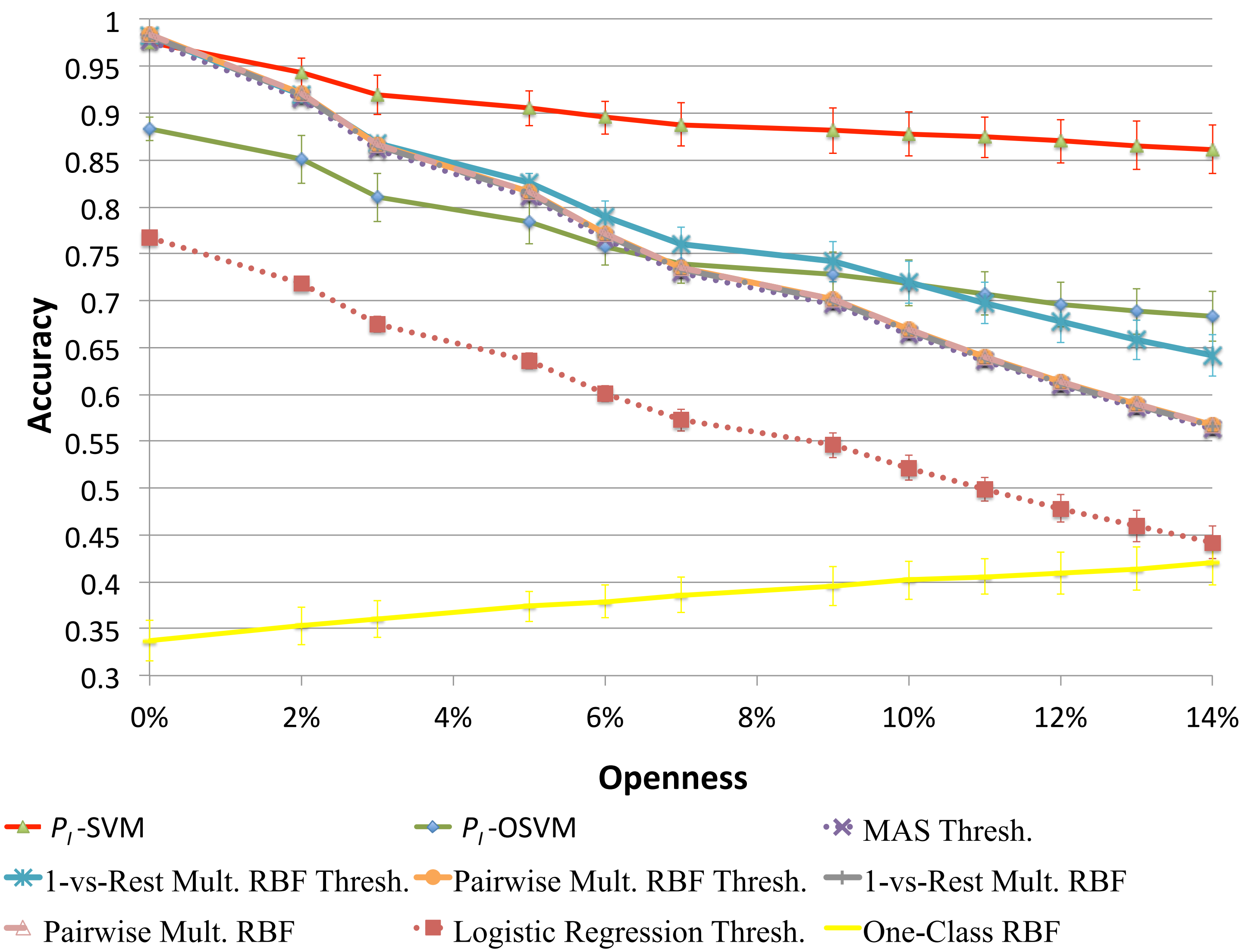
Machine Learning Benchmark: LETTER



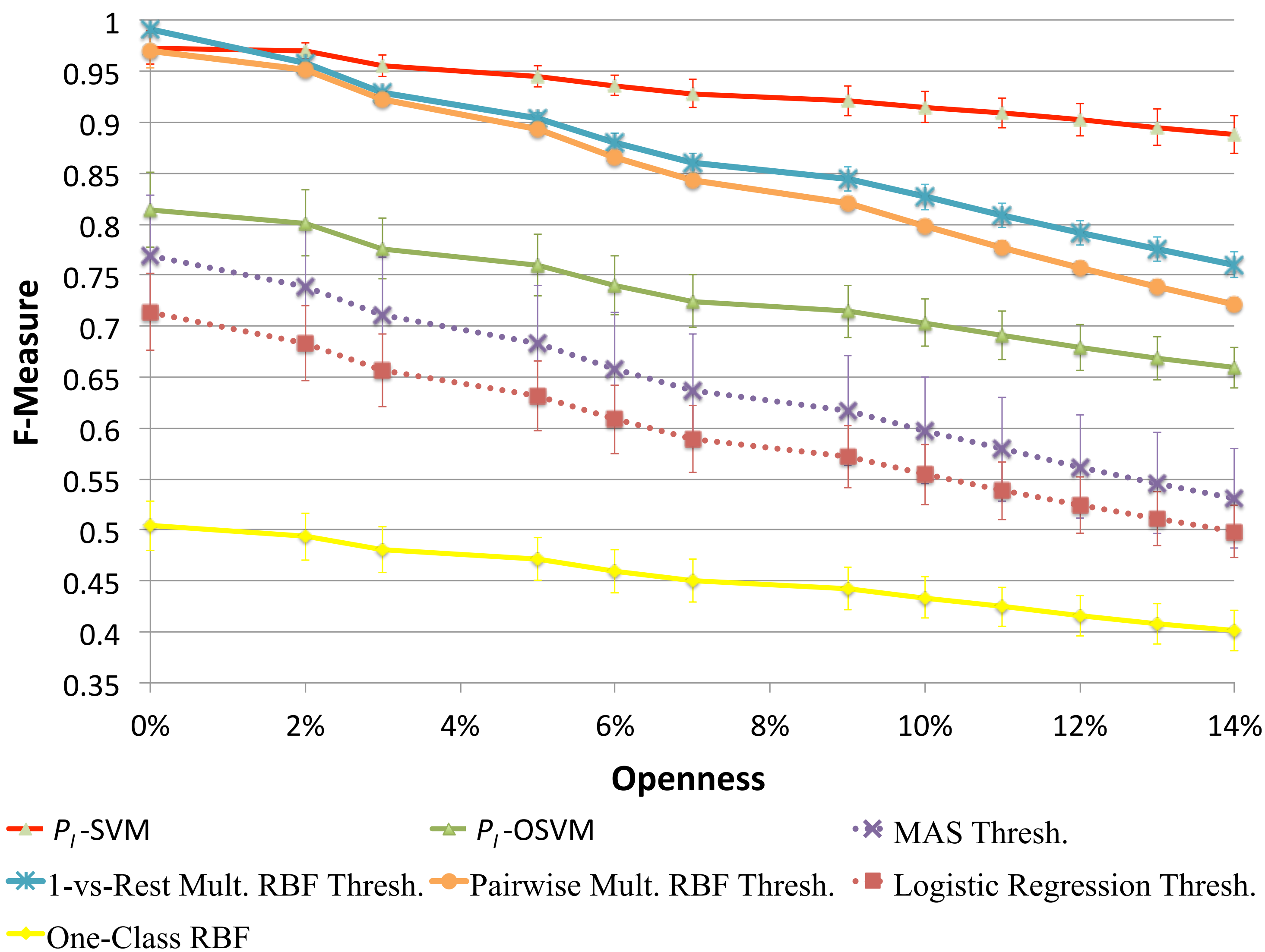
Machine Learning Benchmark: LETTER



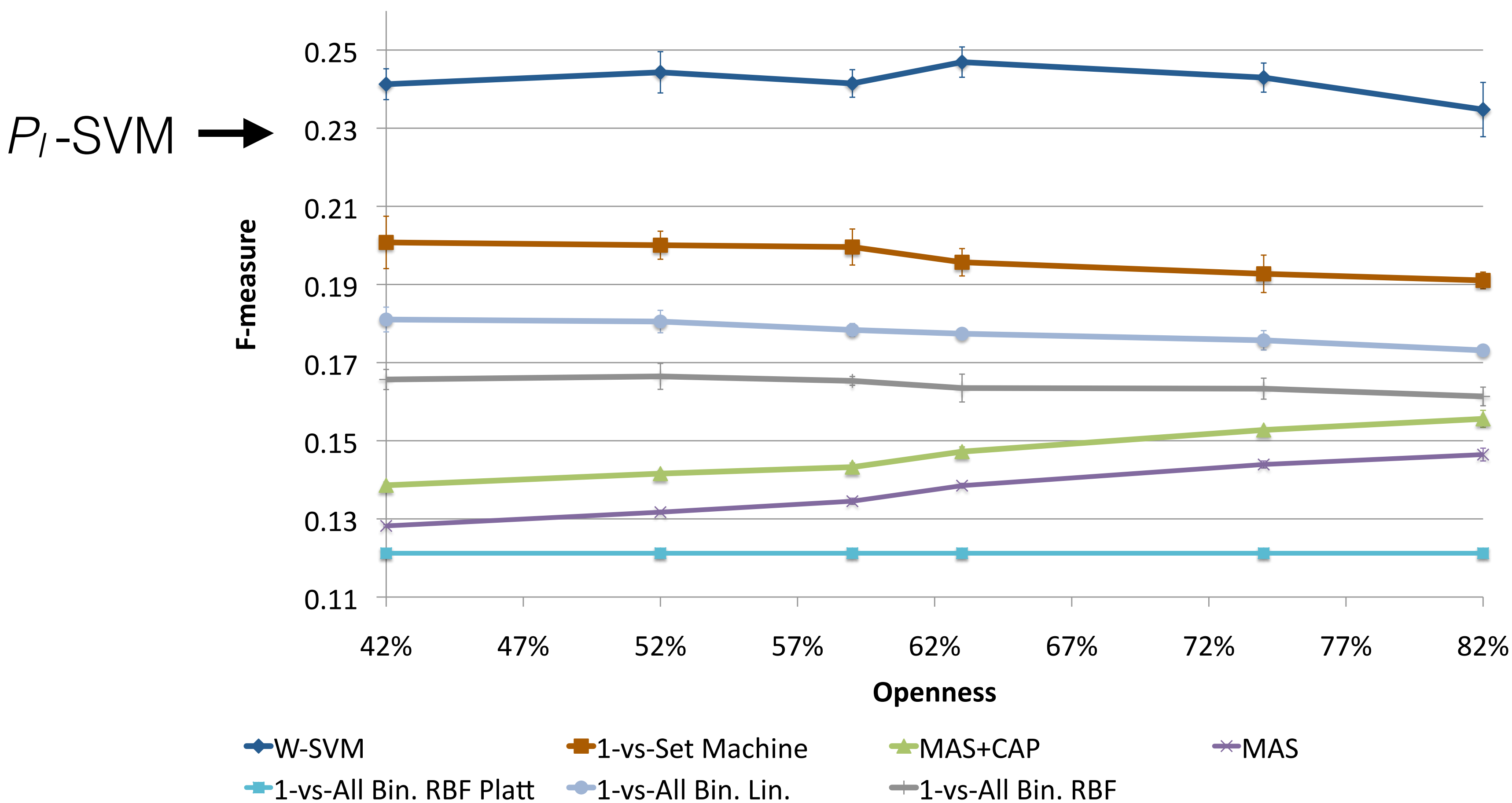
Machine Learning Benchmark: LETTER



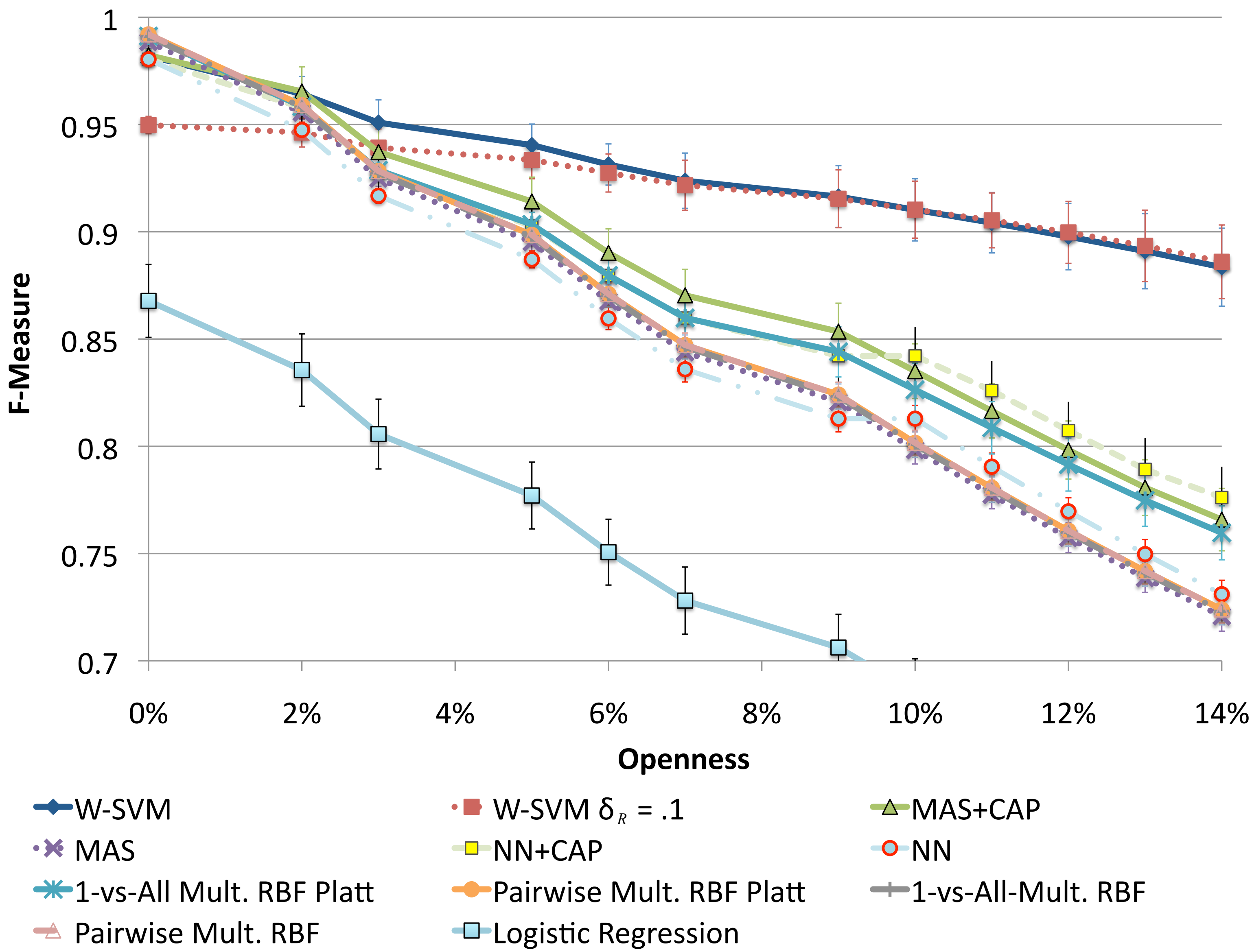
Alternate Priors: Freq. of Occurrence of Letters in a Reference Corpus



W-SVM Object Recognition



Machine Learning Benchmark: LETTER

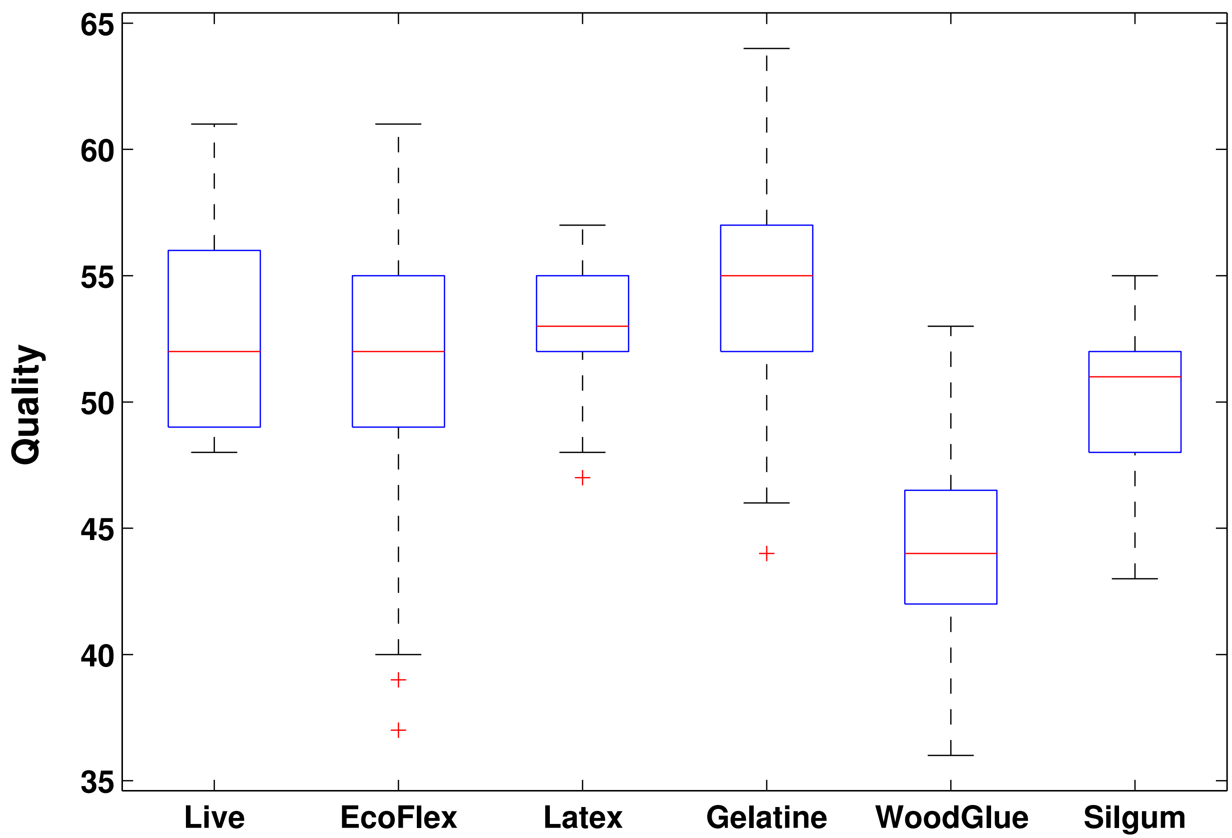


Fingerprint Spoof Detection

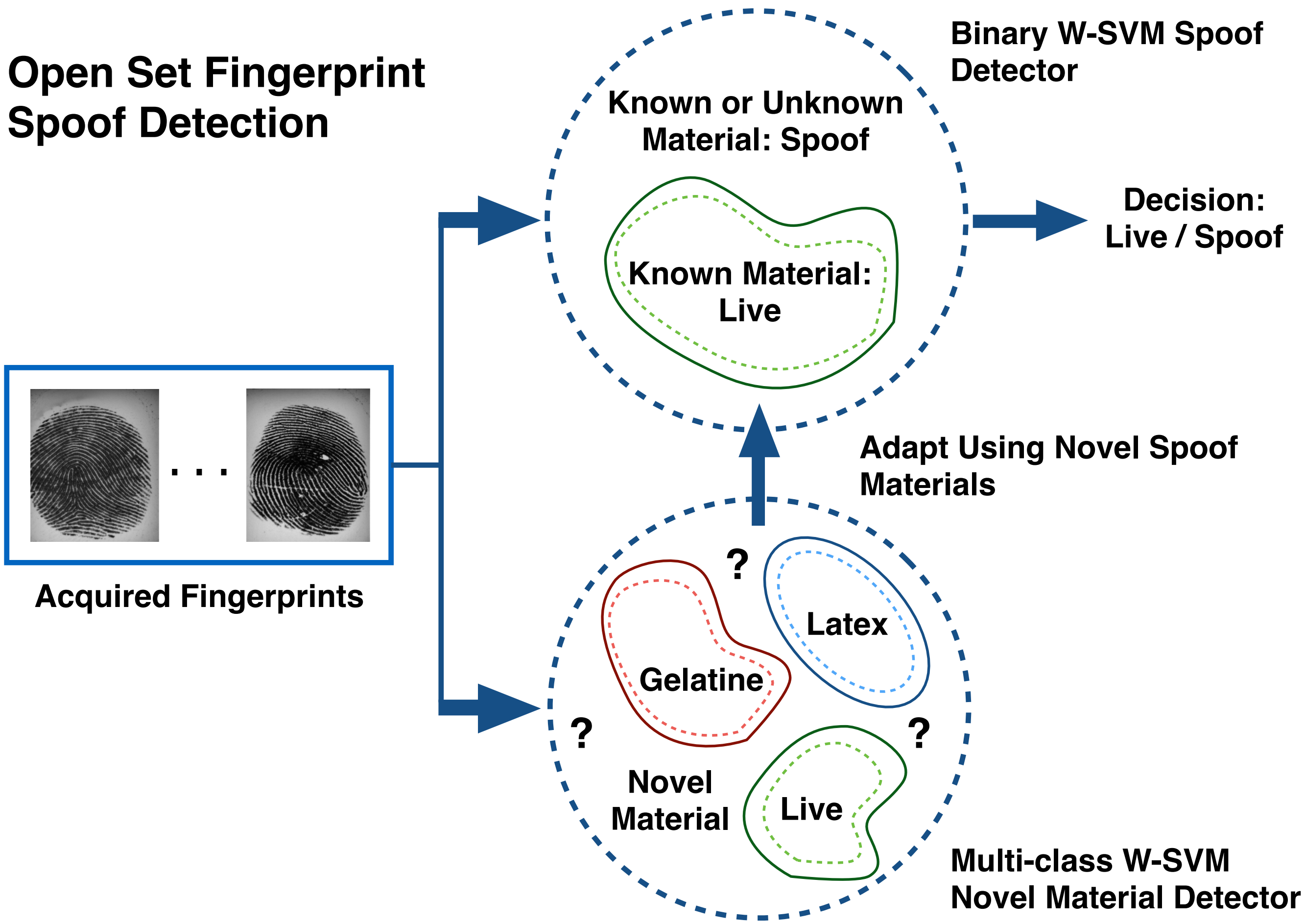
Incomplete knowledge of fabrication materials is always present at training time

(a) EcoFlex (b) Latex (c) Gelatine (d) Silgum (e) WoodGlue

Materials and Quality

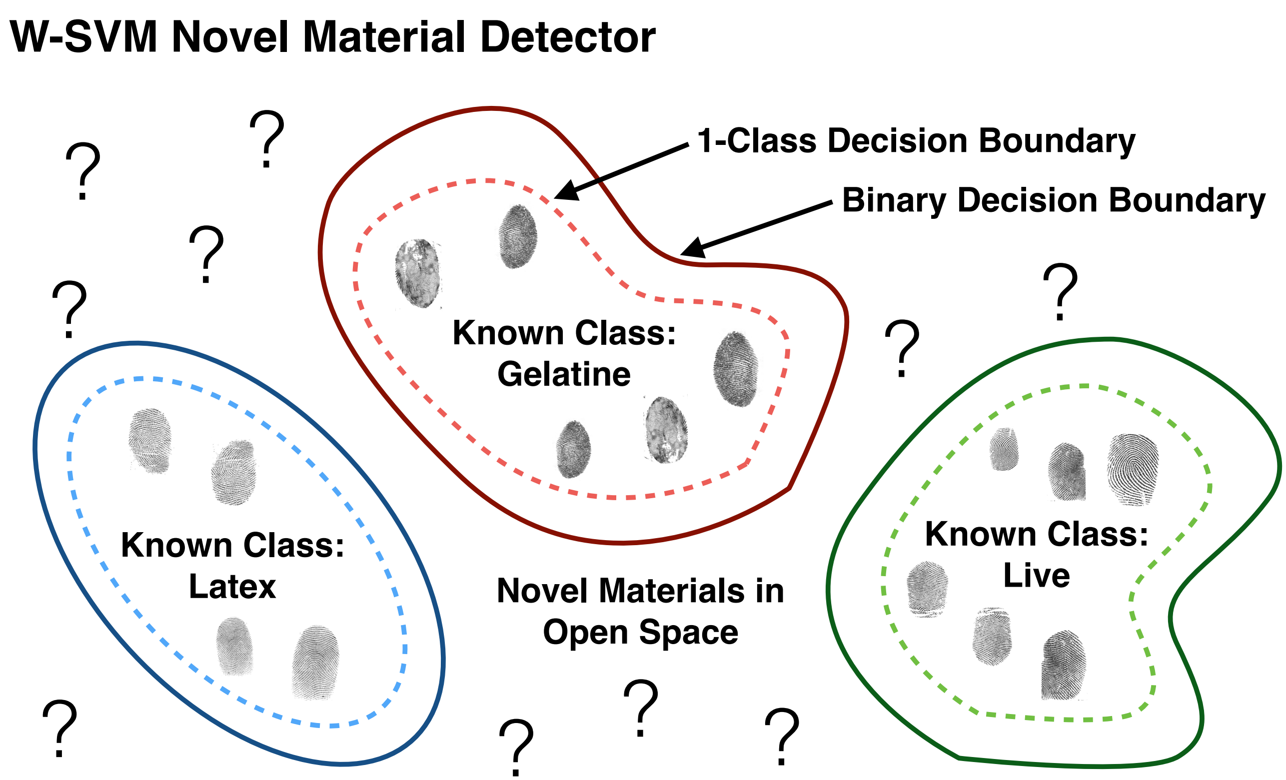


Automatic detection and adaptation of a spoof detector to new spoof materials

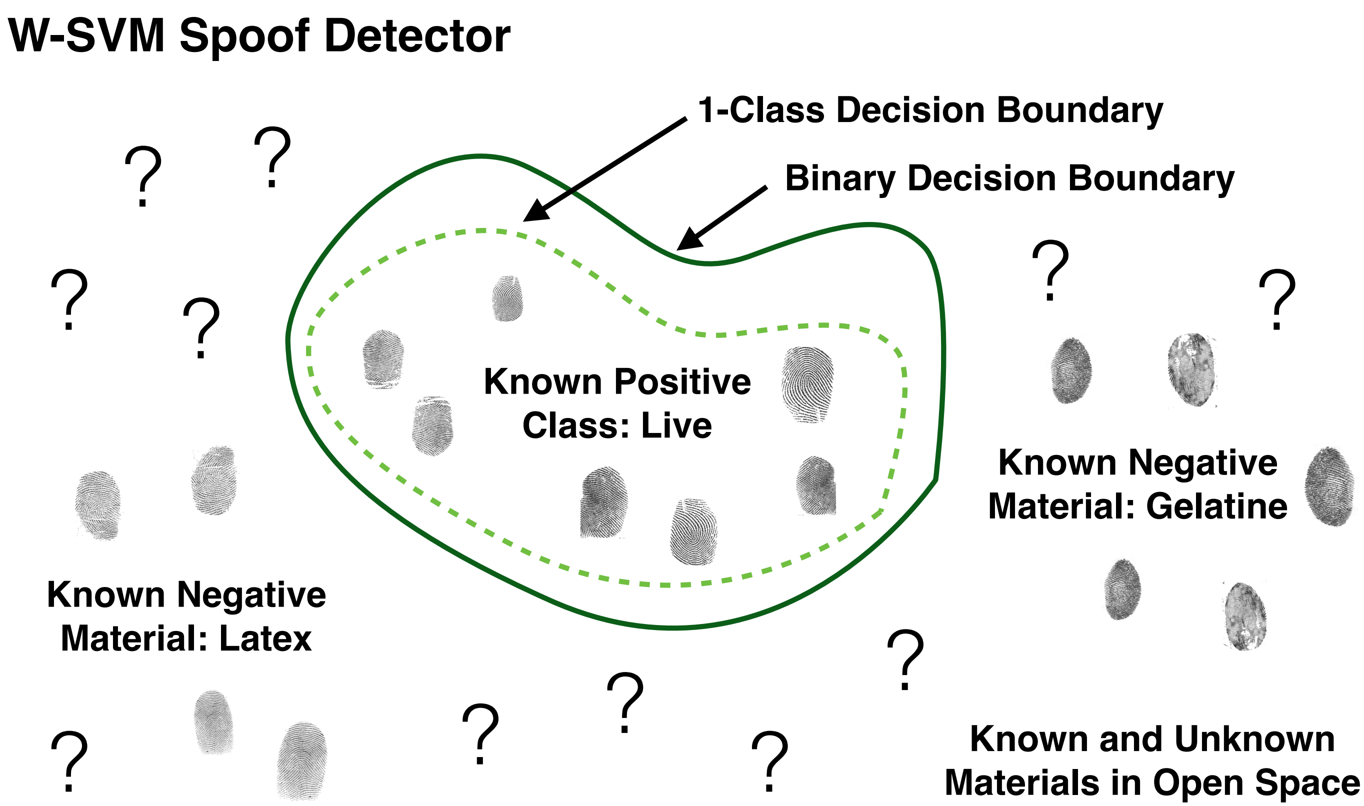


Rattani et al. "Open Set Fingerprint Spoof Detection Across Novel Fabrication Materials, IEEE T-IFS 2015

W-SVM Novel Material Detector



W-SVM Spoof Detector



Experimental assessment of W-SVM

Training: LivDet 2011 is partitioned into 1,000 live and 400 spoof images corresponding to two fabriaction materials

Testing: LivDet 2011 is partitioned into two non-overlapping partitions T_1 and T_2

Each T_i consists of 500 live and 500 spoof images

200 images are from spoof materials known at training time; 300 are from novel materials

http://people.clarkson.edu/projects/biosal/fingerprint/

Performance difference between known and novel materials

Training materials	$\mathcal{L}^{BSIF}$ EER$_{known}$ [%]	$\mathcal{L}^{BSIF}$ EER$_{novel}$ [%]	$\mathcal{L}^{LBP}$ EER$_{known}$ [%]	$\mathcal{L}^{LBP}$ EER$_{novel}$ [%]	$\mathcal{L}^{LPQ}$ EER$_{known}$ [%]	$\mathcal{L}^{LPQ}$ EER$_{novel}$ [%]	Average EER$_{known}$ [%]	Average EER$_{novel}$ [%]
Biometrika								
Skin+Latex+EcoFlex	6.0	16.3	6.5	13.2	9.8	18.4	7.4	16.0
Skin+WoodGlue+Latex	15.0	15.0	10.0	13.8	14.4	16.8	13.1	15.2
Skin+Gelatine+Latex	11.0	16.5	12.0	11.2	8.9	17.7	10.6	15.1
Skin+Silgum+Latex	10.5	20.8	12.3	19.7	10.8	16.3	11.2	18.9
Skin+EcoFlex+Silgum	14.0	29.5	9.3	30.2	12.3	23.0	11.9	27.6
Skin+Gelatine+EcoFlex	13.3	23.3	9.7	15.2	14.0	22.4	12.3	20.3
Skin+Silgum+Gelatine	13.3	23.8	11.5	23.3	14.8	19.5	13.2	22.2
Skin+WoodGlue+Silgum	18.3	23.0	18.0	32.3	13.5	19.0	16.6	24.8
Skin+Gelatine+WoodGlue	16.8	17.2	12.3	11.0	15.8	17.3	15.0	15.2
Skin+WoodGlue+EcoFlex	16.3	17.2	21.7	26.7	17.4	17.3	18.5	20.4
Average EER $\pm$ STDERROR:	13.5 $\pm$ 1.1	**20.3** $\pm$ 1.5	12.3 $\pm$ 1.4	**19.7** $\pm$ 2.5	13.2 $\pm$ 0.9	**18.8** $\pm$ 0.7	12.9 $\pm$ 1.0	**19.6** $\pm$ 1.4

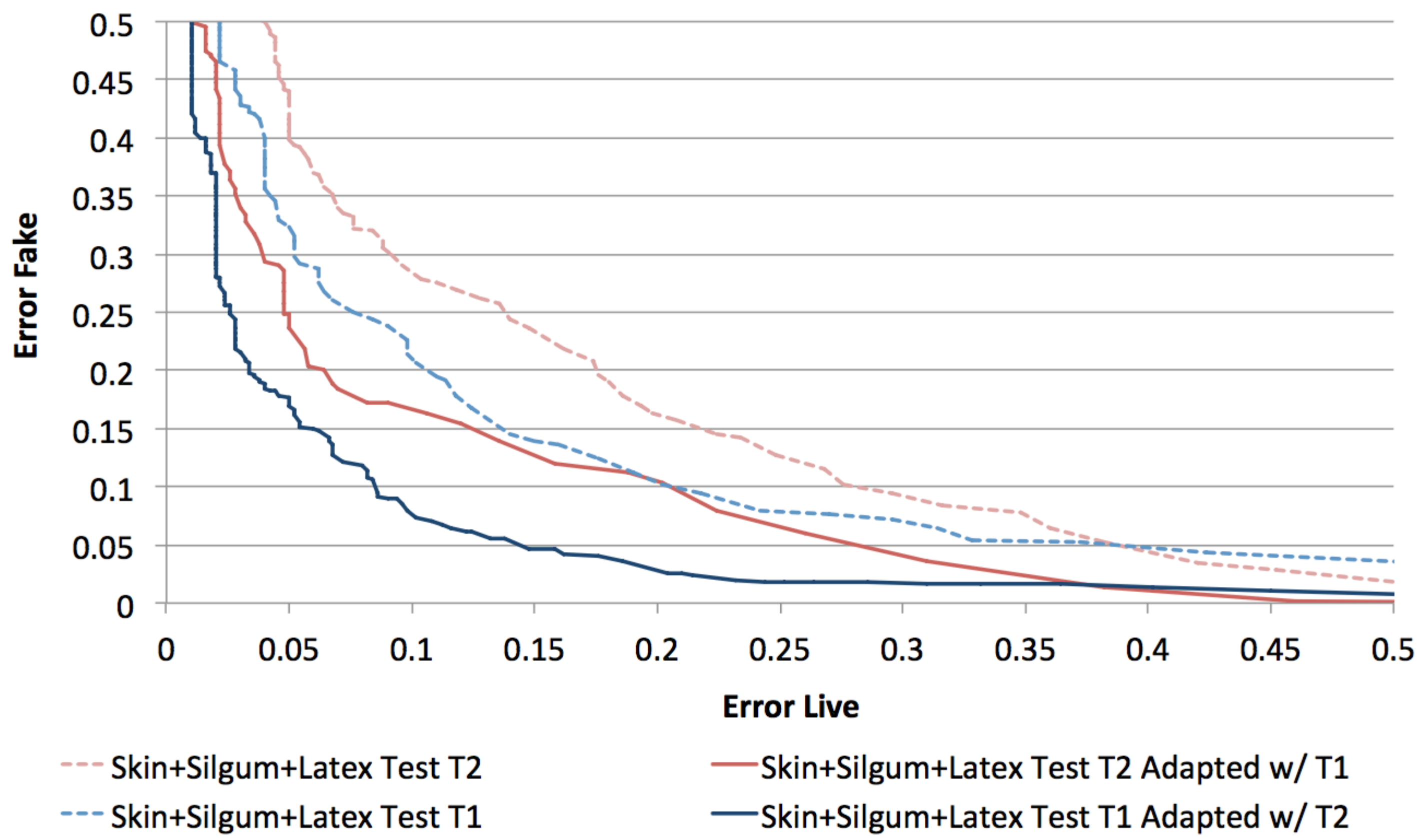
Performance by feature set

Texture descriptors used	$EER_{\mathcal{M}} \pm$ STDERROR [%]			
	Biometrika	Italdata	Digital Persona	Sagem
Grey Level Co-occurence Matrix (GLCM) [16]	44.6 ± 1.7	52.3 ± 2.3	43.7 ± 2.6	43.6 ± 3.4
Binary Statistical Image Features (BSIF) [11]	33.2 ± 1.2	36.9 ± 1.3	34.2± 2.1	38.5± 2.7
Local Phase Quantization (LPQ) [13]	34.3 ± 1.3	36.7 ± 1.4	44.9 ± 5.3	40.3 ± 3.4
Binary Gabor Patterns (BGP) [50]	30.3 ± 1.0	36.8 ± 1.4	34.2 ± 2.3	40.6 ± 2.2
Local Binary Patterns (LBP) [32]	32.5± 2.0	37.3±1.4	36.6± 2.1	31.8± 1.7
Local Binary Patterns (LBP) + Binary Gabor Patterns (BGP)	28.5 ± 1.2	34.1 ± 1.4	31.1 ± 2.3	32.5 ± 2.2

Adapted spoof detector

Training materials	Tested on T_2		Tested on T_1	
	$\mathcal{L}^{LBP}$ (not adapted) [%]	$\mathcal{L}^{LBP'}$ (adapted using T_1) [%]	$\mathcal{L}^{LBP}$ (not adapted) [%]	$\mathcal{L}^{LBP'}$ (adapted using T_2) [%]
Skin+Latex+EcoFlex	14.6	13.4	7.0	5.0
Skin+WoodGlue+Latex	12.8	9.6	9.8	6.0
Skin+Gelatine+Latex	13.8	13.4	10.2	7.8
Skin+Silgum+Latex	18.2	14.0	14.2	9.0
Skin+EcoFlex+Silgum	29.6	18.0	21.0	9.0
Skin+Gelatine+EcoFlex	15.2	14.2	10.4	7.2
Skin+Silgum+Gelatine	22.2	15.8	18.2	10.0
Skin+WoodGlue+Silgum	30.4	14.4	27.2	9.2
Skin+Gelatine+WoodGlue	12.2	10.8	10.0	8.2
Skin+WoodGlue+EcoFlex	19.8	12.8	12.2	6.0
Average EER $\pm$ STDERROR :	18.9 $\pm$ 2.1	**13.6** $\pm$ 0.7	14.0 $\pm$ 2.0	**7.7** $\pm$ 0.5

DET curves shift to the left after adaptation

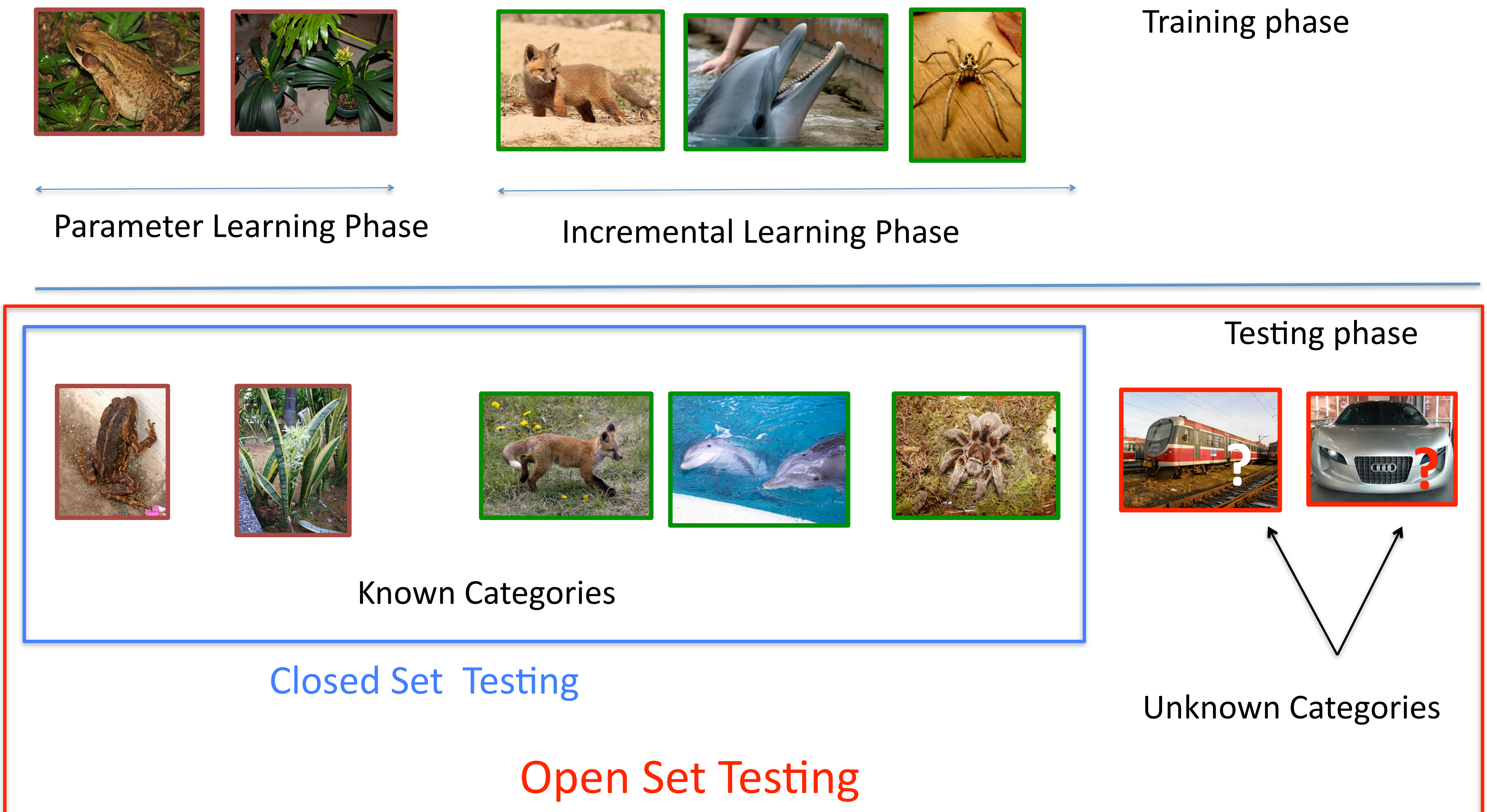


Better

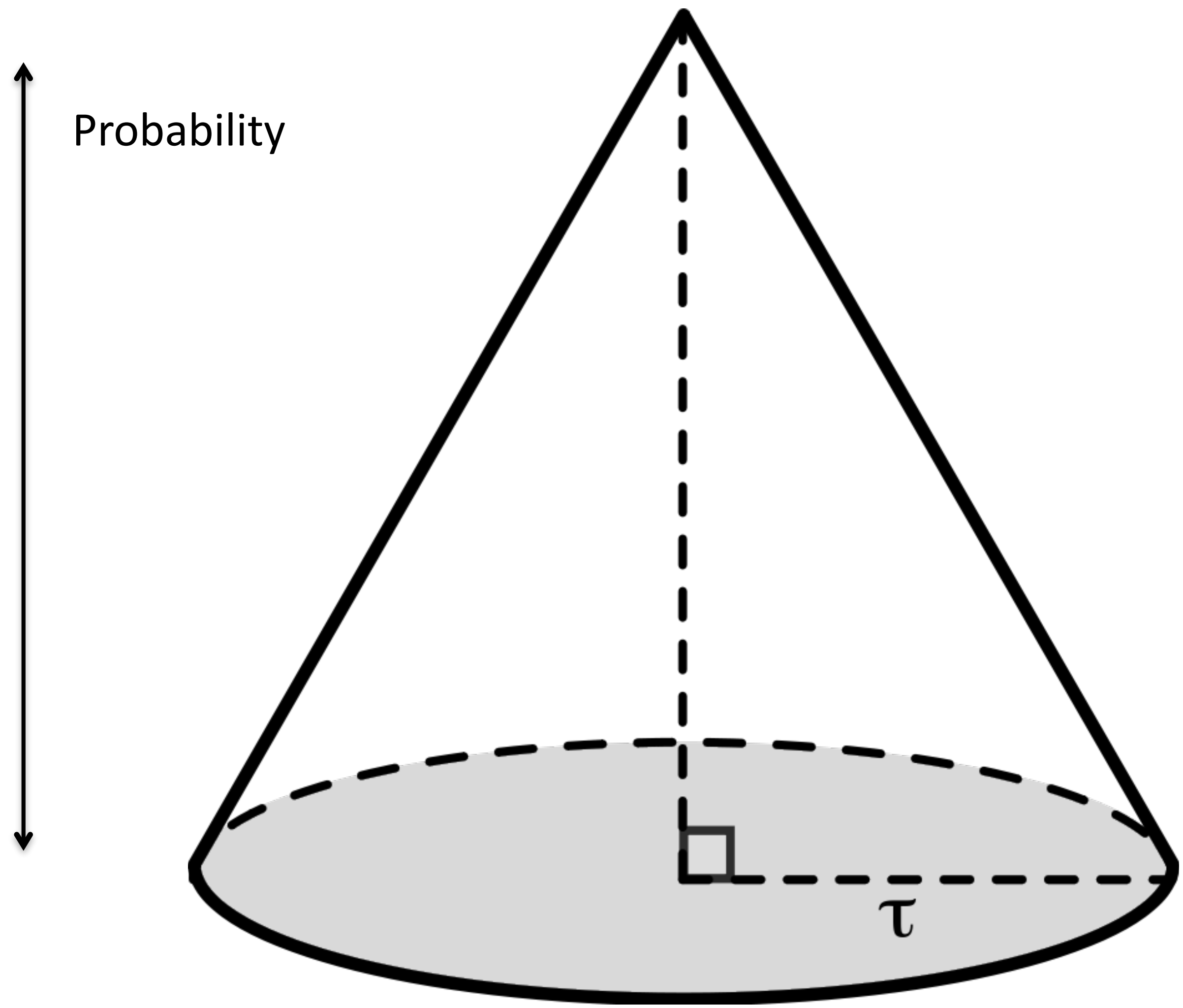
How well could you do with these features and the W-SVM?

Sensors	Tested on T_2 (not adapted) [%]	Tested on T_2 (adapted using T_1) [%]	Tested on T_1 (not adapted) [%]	Tested on T_1 (adapted using T_2) [%]
Biometrika				
	$\mathcal{L}^{LBP}$	$\mathcal{L}^{LBP'}$	$\mathcal{L}^{LBP}$	$\mathcal{L}^{LBP'}$
Average EER STDERROR :	18.9 $\pm$ 2.1	**13.5** $\pm$ 0.6	14.0 $\pm$ 2.0	**7.7** $\pm$ 0.4
	$\mathcal{L}^{LPQ}$	$\mathcal{L}^{LPQ'}$	$\mathcal{L}^{LPQ}$	$\mathcal{L}^{LPQ'}$
Average EER $\pm$ STDERROR:	20.3 $\pm$ 0.5	**14.6** $\pm$ 0.5	12.5 $\pm$ 0.7	**9.0** $\pm$ 0.5
	$\mathcal{L}^{BSIF}$	$\mathcal{L}^{BSIF'}$	$\mathcal{L}^{BSIF}$	$\mathcal{L}^{BSIF'}$
Average EER $\pm$ STDERROR:	21.5 $\pm$ 1.3	**15.4** $\pm$ 0.6	13.1 $\pm$ 0.9	**7.0** $\pm$ 0.4

Open World Evaluation

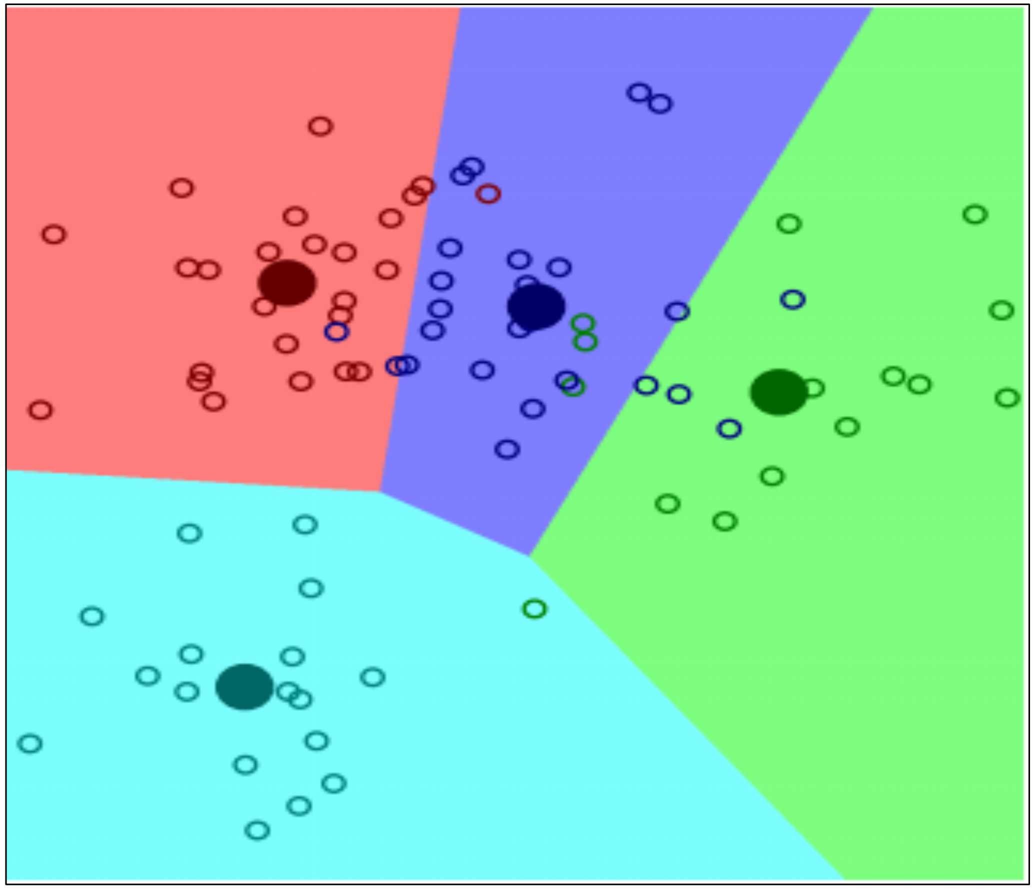


Opening an Existing Algorithm: Nearest Non-Outlier (NNO) Algorithm



A. Bendale, T. Boult "Towards Open World Recognition" CVPR 2015

NCM – Metric Learning



NCM Classifier with Metric Learning

T Mensink, J Verbeek, F Perronin, G Csurka “Distance based Image Classification: Generalizing to New Classes at Near Zero Cost” IEEE TPAMI 2013

M Ristin, M Guillaumin, J Gall, L Van Gool “Incremental Learning of NCM Forests for Large-Scale Image Classification” CVPR 2014

Opening an Existing Algorithm: Nearest Non-Outlier (NNO) Algorithm

Standard gamma function
In volume of m-D ball

τ is threshold for open world

Class mean for class i

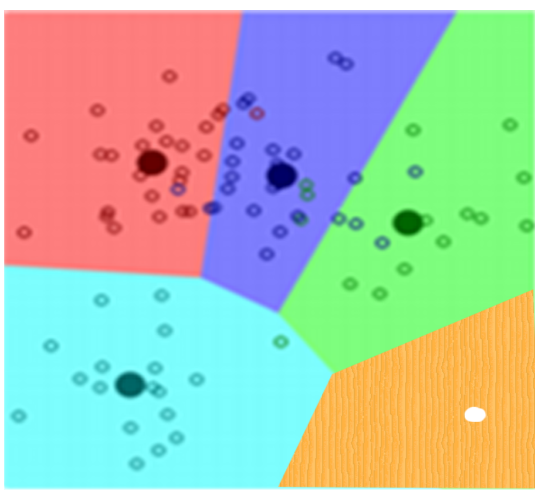
$$\hat{f}_i(x) = \frac{\Gamma(\frac{m}{2}+1)}{\pi^{\frac{m}{2}}\tau^m}\left(1 - \frac{1}{\tau}\|W^\top x - W^\top \mu_i\|\right) \qquad (1)$$

be our measurable recognition function with $\hat{f}_i(x) > 0$ giving the probability of being in class i.

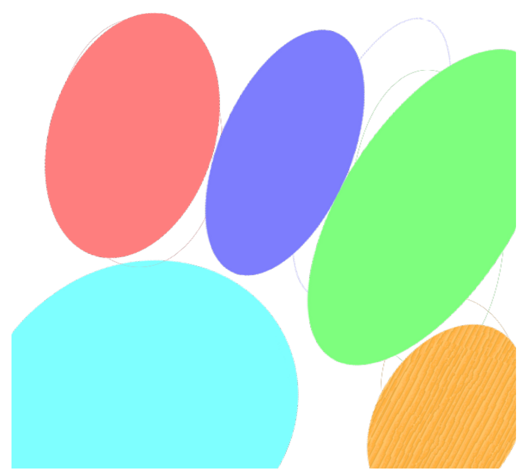
W = Linear Transformation (weight matrix from metric learning)

Training for Open World

- Parameter Learning with initial set of categories
- Estimation of τ for open set learning to balance open space risk
- Optimize for Known vs Unknown Errors
- Incrementally add new categories

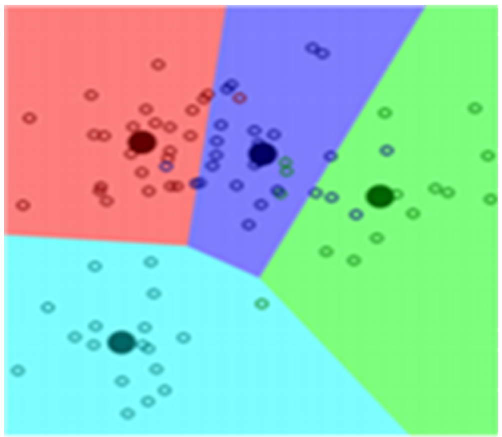


NCM - ML

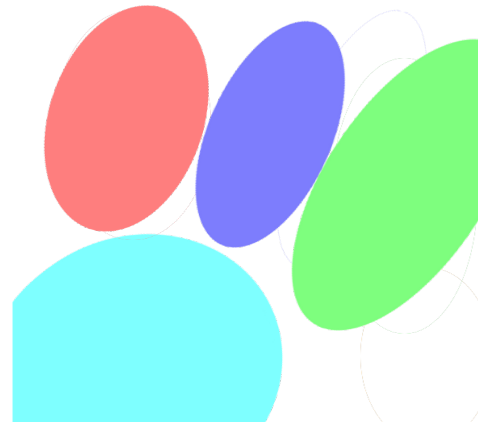


NNO

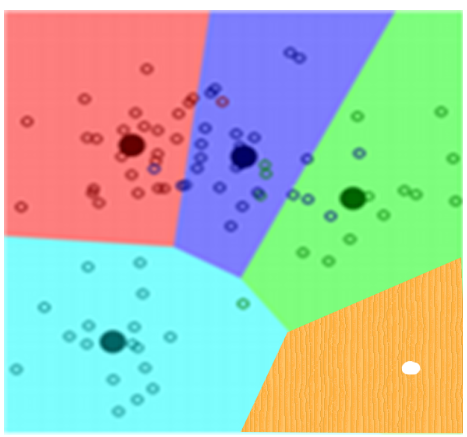
Learning Novel Concepts

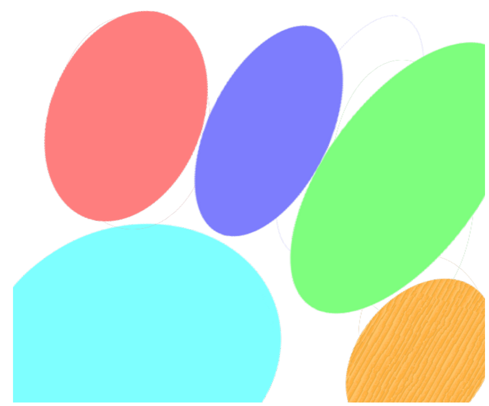


Nearest Class Mean Classifier



Nearest Non Outlier Algorithm





Adding Novel Concepts to the System

Experiments

Datasets

- ILSVRC'10: 1.2M training images, 1000 classes
- ILSVRC'12: 1.2M training images, 1000 classes

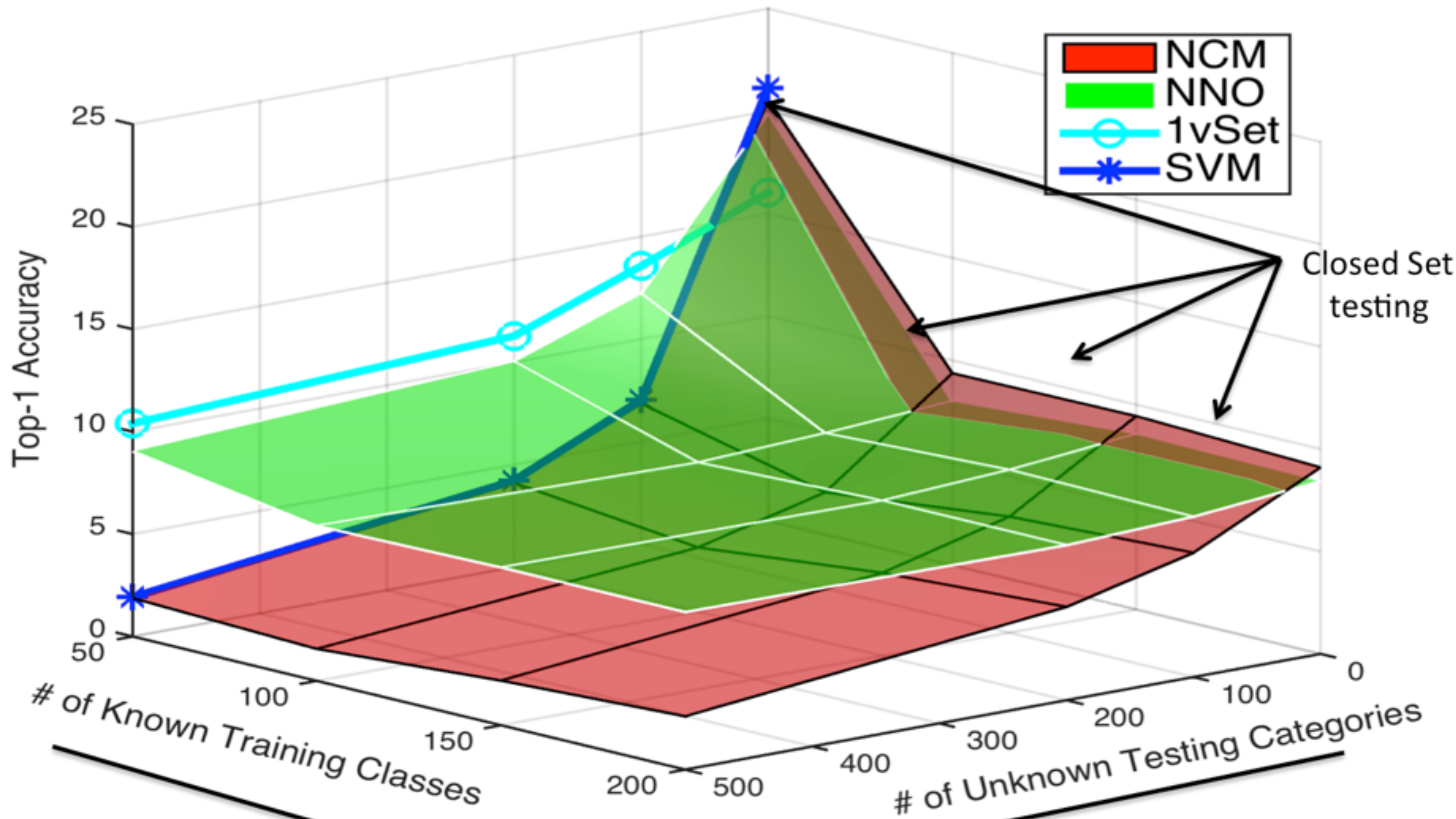
Features

- Dense SIFT features, Quantized into 1000 Bag of Visual Words
- Publically available features
- LBP, HOG, Dense SIFT (for ILSVRC'12)

Algorithms

- Nearest Class Mean - ML Classifier (NCM) [*Mensink etal PAMI 2013*]
- Nearest Non-Outlier Algorithm (NNO) [*This Paper*]
- 1vSet [*Scheirer etal PAMI 2013*]
- Linear SVM [*Liblinear, Fan etal JMLR 2008*]

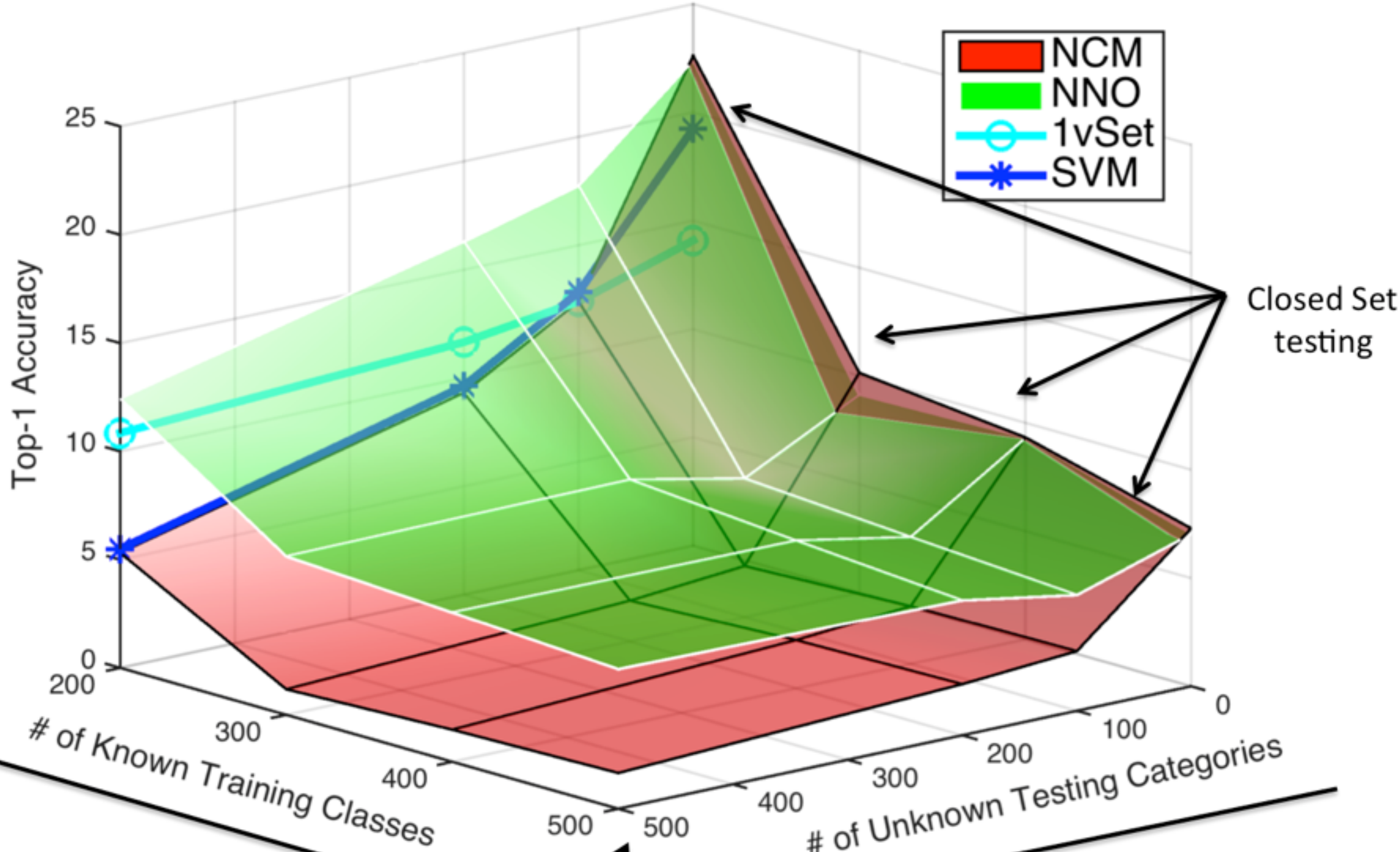
50 Initial Categories



Incrementally adding categories during training

Increasing # of unknown categories during testing i.e. increasing openness of problem

200 Initial Categories



Incrementally adding categories during training

500 known +
500 unknown categories

Increasing # of unknown categories during testing i.e. increasing openness of problem

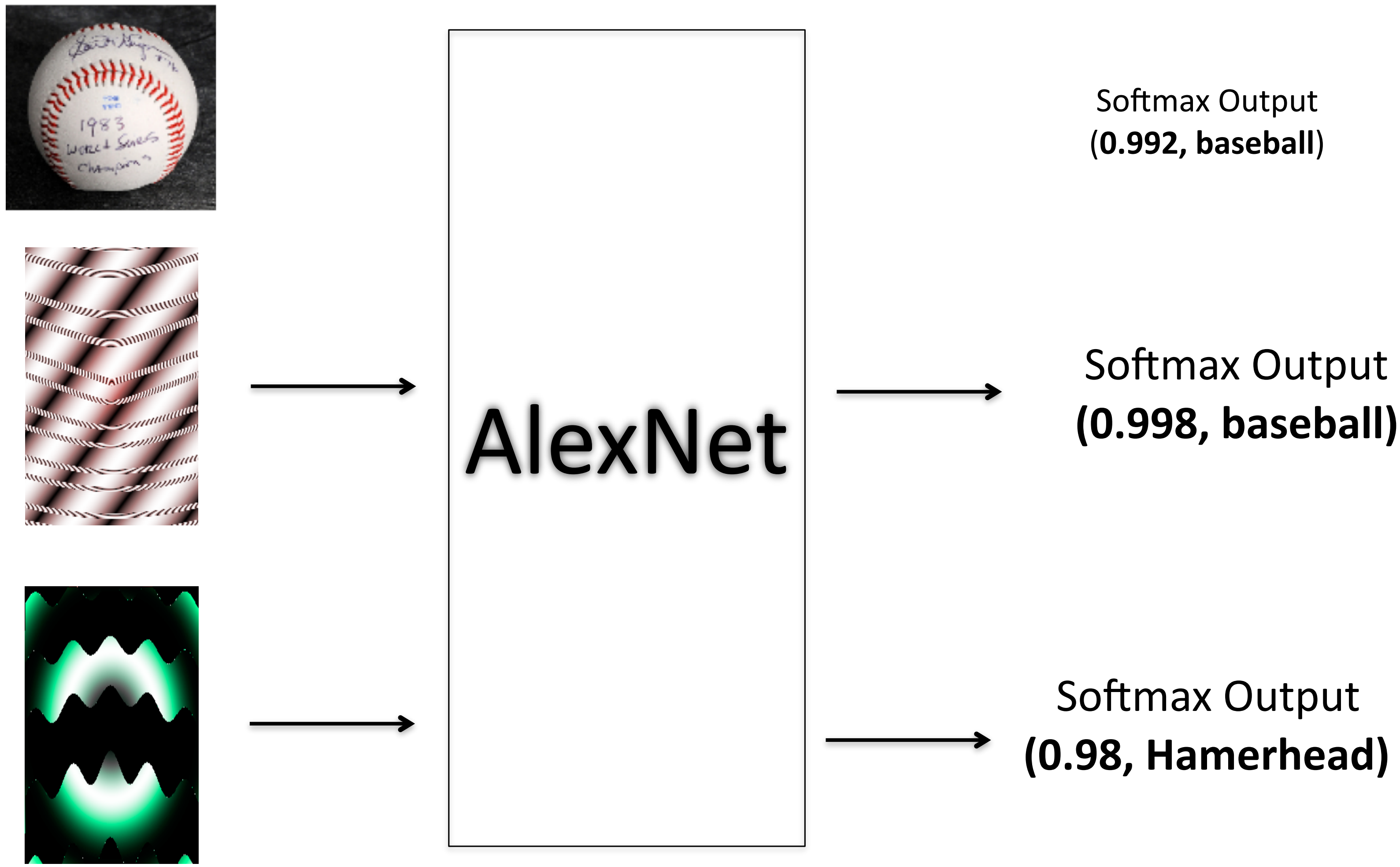
Opening Deep Networks

- Softmax always has a “winner” and re-weights scores
- Networks are easily fooled with high confidence
- “Fooling” images are obviously “open set” and should be rejected
- Adversarial images are more problematic - visually close but often far in label space

A. Bendale and T. Boult “Towards Open Set Deep Networks” CVPR 2016 (Short oral)

Opening Deep Networks

Can hill climb to find fooling images*



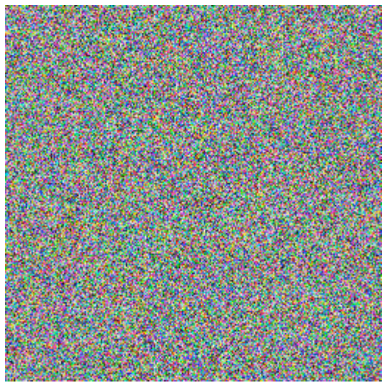
* A. Nguyen, J. Yosinski, and J. Clune "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images" CVPR 2015

Adversarial Manipulation of AlexNet

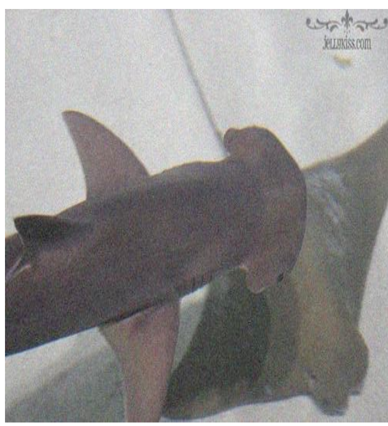
Hammerhead Image

+

Noise (*100)



These are “visually near” but mislabeled



AlexNet

Softmax Output
(.32, **ScubaDiver**)

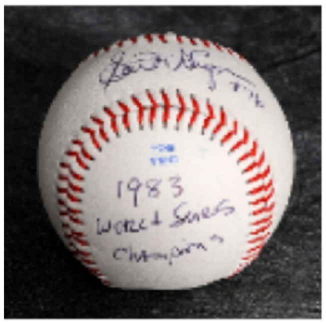
Adversarial images generated using: Goodfellow, Shelns and Szegedy “Explaining and harnessing adversarial examples,” ICLR 2015

MAV and OpenMax

- Insight: A class is represented not just by its output, but by its Mean Activation Layer (scores for all classes)
- MAV is just the average in penultimate layer
- “EVT distances” from MAV is a CAP model
- Given MAV, estimate probability of “unknown” via EVT and OpenMax = Softmax type normalized probability including probability of unknown

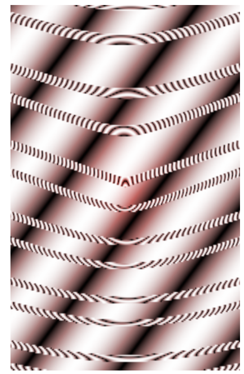
Open Set Deep Networks

Idealized class

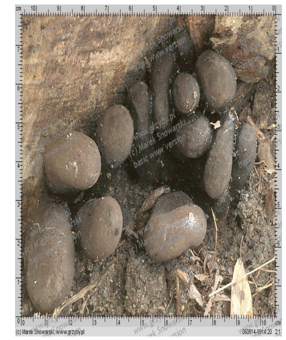


Softmax Output
(**0.992, baseball**)

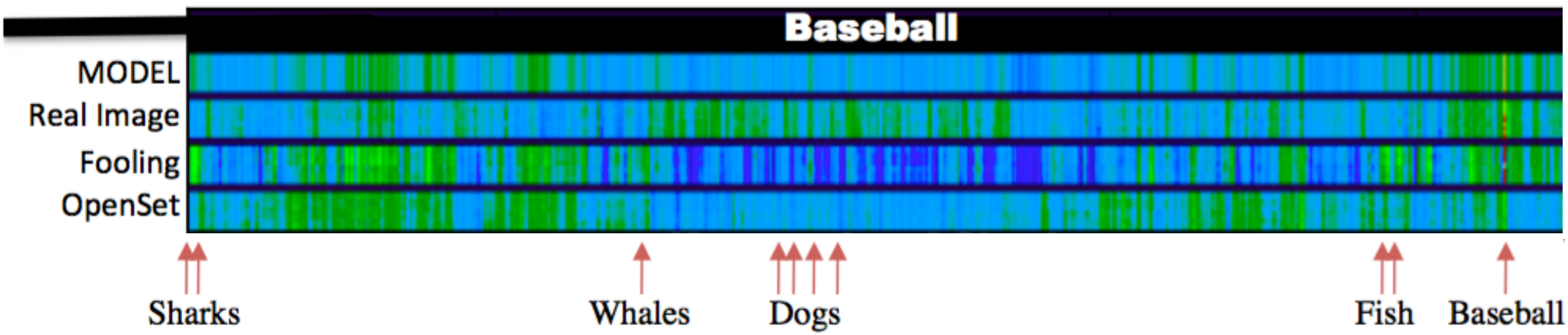
Real: SM 0.94



Fooling: SM 1.0,



Openset: SM 0.15



Step 1: Represent "known" as mean activation of a class + EVT-model for "outlier"

Algorithm 1 EVT Meta-Recognition Calibration for Open Set Deep Networks, with per class Weibull fit to η largest distance to mean activation vector. Returns libMR models ρ_j which includes parameters τ_i for shifting the data as well as the Weibull shape and scale parameters: κ_i, λ_i.

Require: FitHigh function from libMR

Require: Activation levels in the penultimate network layer $\mathbf{v}(\mathbf{x}) = v_1(x) \ldots v_N(x)$

Require: For each class j let $S_{i,j} = v_j(x_{i,j})$ for each correctly classified training example $x_{i,j}$.

1: **for** $j = 1 \ldots N$ **do**

2: **Compute mean AV**, $\mu_j = mean_i(S_{i,j})$

3: **EVT Fit** $\rho_j = (\tau_j, \kappa_j, \lambda_j) = \text{FitHigh}(\|\hat{S}_j - \mu_j\|, \eta)$

4: **end for**

5: **Return** means μ_j and libMR models ρ_j

Step 2: Compute "open max" with explicit probably of unknown

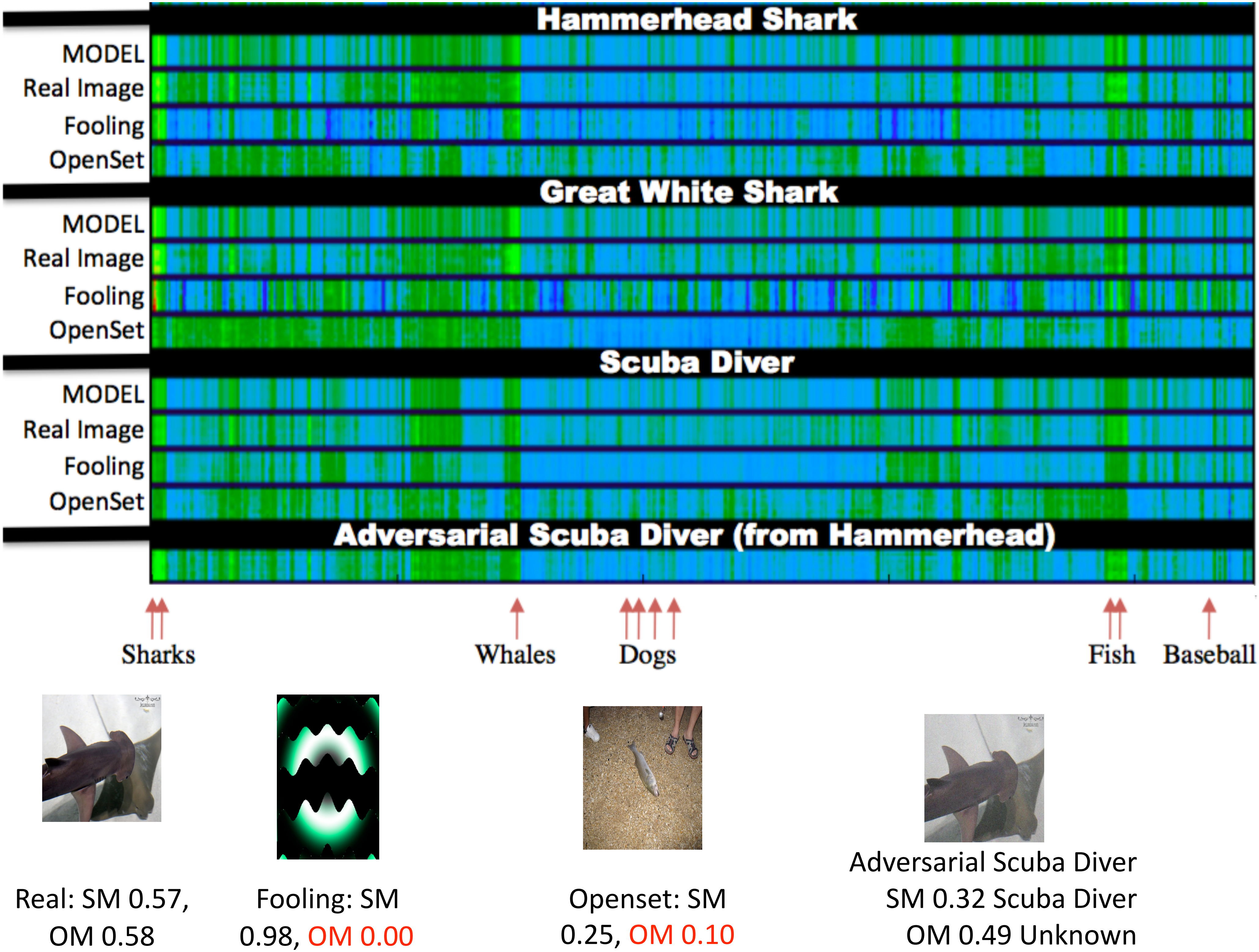
Algorithm 2 OpenMax probability estimation with rejection of unknown or uncertain inputs.

Require: Activation vector for $\mathbf{v}(\mathbf{x}) = v_1(x), \ldots, v_N(x)$
Require: **means** μ_j and libMR models $\rho_j = (\tau_i, \lambda_i, \kappa_i)$
Require: α, the numer of "top" classes to revise

1: Let $s(i) = \text{argsort}(v_j(x))$; Let $\omega_j = 1$
2: **for** $i = 1, \ldots, \alpha$ **do**
3: $\quad \omega_{s(i)}(x) = 1 - \frac{\alpha - i}{\alpha} e^{-\left(\frac{\|x - \tau_{s(i)}\|}{\lambda_{s(i)}}\right)^{\kappa_{s(i)}}}$
4: **end for**
5: Revise activation vector $\hat{v}(x) = \mathbf{v}(\mathbf{x}) \circ \omega(\mathbf{x})$
6: Define $\hat{v}_0(x) = \sum_i v_i(x)(1 - \omega_i(x))$.
7:

$$\hat{P}(y = j|\mathbf{x}) = \frac{e^{\hat{\mathbf{v}}_{\mathbf{j}}(\mathbf{x})}}{\sum_{i=0}^{N} e^{\hat{\mathbf{v}}_{\mathbf{i}}(\mathbf{x})}} \tag{2}$$

8: Let $y^* = \text{argmax}_j \, P(y = j|\mathbf{x})$
9: Reject input if $y^* == 0$ or $P(y = y^*|\mathbf{x}) < \epsilon$

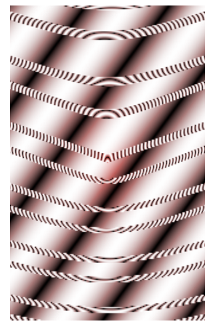


Hammerhead Shark
MODEL
Real Image
Fooling
OpenSet
Great White Shark
MODEL
Real Image
Fooling
OpenSet
Scuba Diver
MODEL
Real Image
Fooling
OpenSet
Adversarial Scuba Diver (from Hammerhead)
Sharks
Whales
Dogs
Fish
Baseball
Real: SM 0.57, OM 0.58
Fooling: SM 0.98, OM 0.00
Openset: SM 0.25, OM 0.10
Adversarial Scuba Diver SM 0.32 Scuba Diver OM 0.49 Unknown

Open Set Deep Networks

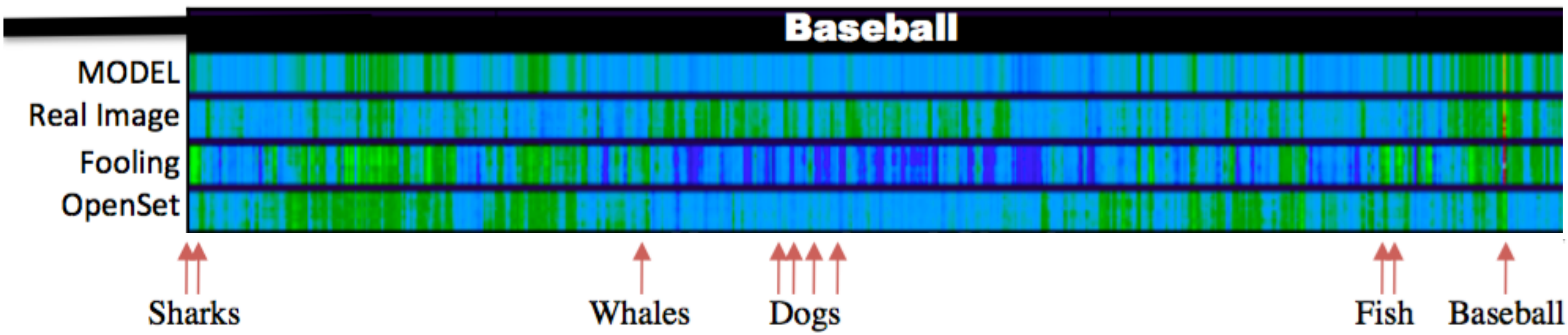
Text

Real: SM 0.94
OM 0.94



Fooling: SM 1.0,
OM 0.00

Openset: 0.15,
OM: 0.17



Wrapping up…

Further Reading

- F. Costa, E. Silva, M. Eckmann, W.J. Scheirer, and A. Rocha, “Open Set Source Camera Attribution and Device Linking,” Pattern Recognition Letters, 2014.
- W.J. Scheirer, A. Rocha, A. Sapkota, and T. Boult, “Towards Open Set Recognition,” IEEE T-PAMI, 35(7) July 2013.
- M.J. Wilber, W.J. Scheirer, P. Leitner, B. Heflin, J. Zott, D. Reinke, D. Delaney, T.E. Boult, “Animal Recognition in the Mojave Desert: Vision Tools for Field Biologists,” IEEE WACV, 2013.
- B. Heflin, W.J. Scheirer, and T.E. Boult, “Detecting and Classifying Scars, Marks, and Tattoos Found in the Wild,” IEEE BTAS, 2012.
- W.J. Scheirer, A. Rocha, R. Micheals, and T.E. Boult, “Meta-Recognition: The Theory and Practice of Recognition Score Analysis,” IEEE T-PAMI, 33(8), 2011.

Further Reading

- A. Rattani, W.J. Scheirer, and A. Ross, “Open Set Fingerprint Spoof Detection Across Novel Fabrication Materials,” IEEE T-IFS, 10(11) Nov. 2015.
- W.J. Scheirer, L.P. Jain, and T.E. Boult, “Probability Models for Open Set Recognition,” IEEE T-PAMI, 36(11), Nov. 2014.
- L.P. Jain, W.J. Scheirer, and T.E. Boult, “Multi-class Open Set Recognition Using Probability of Inclusion,” ECCV, Sept. 2014.

Code

1-vs-Set Machine, P_I-SVM, and W-SVM on GitHub:
https://github.com/ljain2/libsvm-openset

Data sets:
http://www.metarecognition.com/openset/