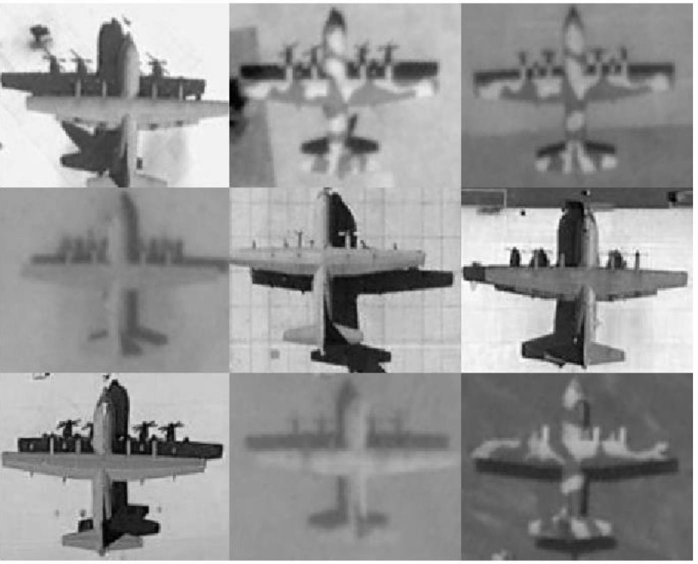
SAND2017-3741C



Open Set Recognition of Aircraft in Aerial Imagery using Synthetic Template Models

Aleksander Bapst, Jonathan Tran, Mark W. Koch,
Mary M. Moya, and Robert Swahn

SPIE Defense + Security 2017

04/10/17

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

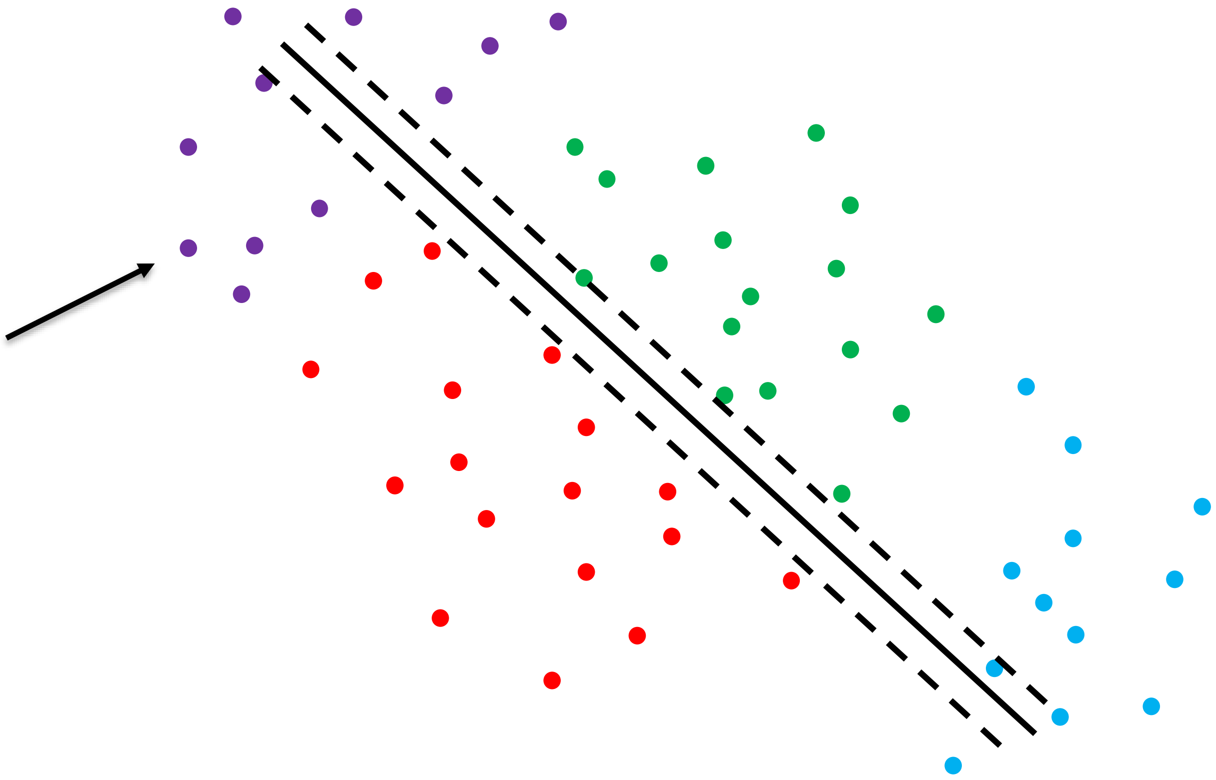
Sandia National Laboratories is a multi-mission laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

Machine learning is at a crossroads.

- High-risk applications demand a high standard of reliability
  - Defense and surveillance
  - Autonomous driving
- Robust decision-making requires robust evidence
- Existing benchmarks treat classification and recognition as a closed-set problem
- Also require VERY large annotated training sets
  - Pascal VOC
  - ImageNet
  - CoCo

Limitations of supervised algorithms

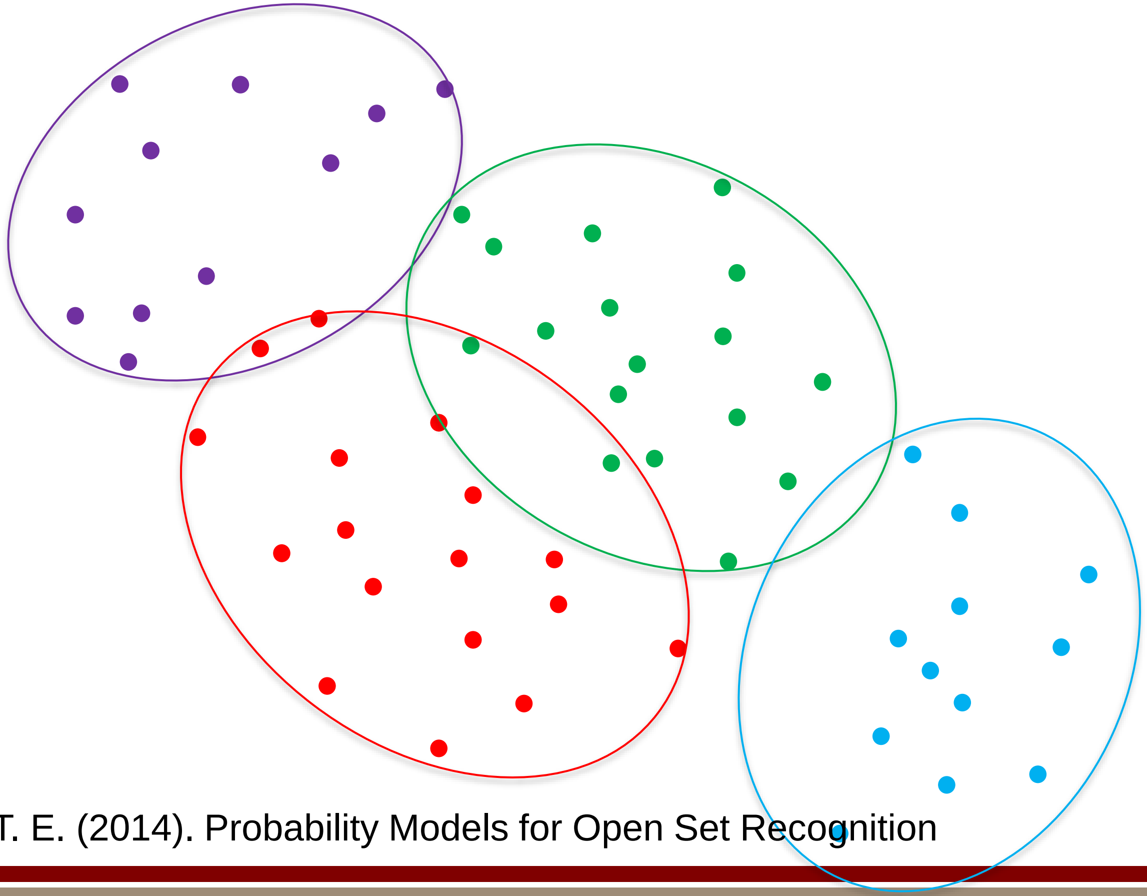
- *Closed set*: learn to separate classes based on modelling of inter-class differences rather than intra-class similarities
- The space of unmodeled data is infinitely large yet isn't addressed at all (open set risk).



Should be classified as "unknown", not red or green.

Limitations of supervised algorithms

- We wish to define classes based on intra-class similarity
- Existing approaches based on "closeness" to a class are generally unreliable[1]
  - One-class SVM
  - Clustering
  - Nearest-neighbor



[1] Scheirer, W. J., Jain, L. P. and Boult, T. E. (2014). Probability Models for Open Set Recognition

Research Goals

- Introduce robust rejection criteria to supervised learning algorithms, i.e. “unknown” classification
- Can we reduce the data requirements for training using more discriminative algorithms and/or synthetic data?
  - More on this in a bit…

Types of data

(1) Known (target) classes: data that are labeled as positive training examples

(2) Known unknowns (non-target): data that are labeled as negative training examples

(3) Unknown unknowns (non-target): classes that are not seen during training

Scheirer, W. J., Jain, L. P. and Boult, T. E. (2014). Probability Models for Open Set Recognition

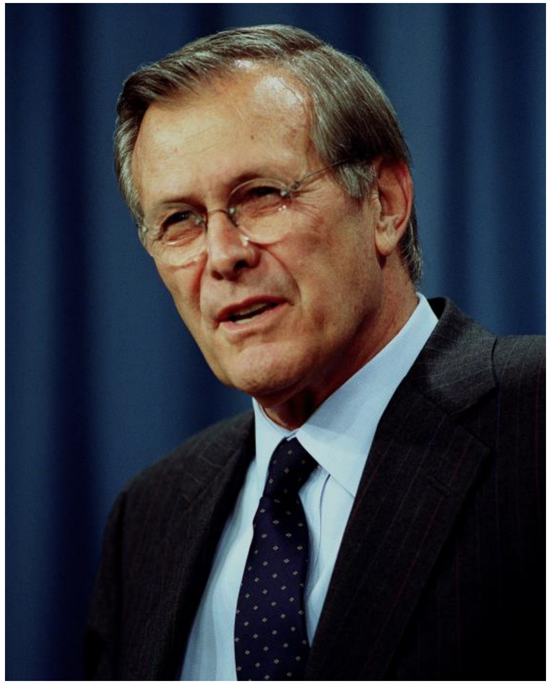
Types of data

(1) Known (target) classes: data that are labeled as positive training examples

(2) Known unknowns (non-target): data that are labeled as negative training examples

(3) Unknown unknowns (non-target): classes that are not seen during training

Terminology inspired by language used by Donald Rumsfeld in 2002



Scheirer, W. J., Jain, L. P. and Boult, T. E. (2014). Probability Models for Open Set Recognition

Open set recognition of aircraft

Goal: recognize specific aircraft models in aerial imagery

Criteria:

- Require strong rejection of non-targets.
- Would also like to use synthetic data to avoid expensive data collection.

Algorithms:

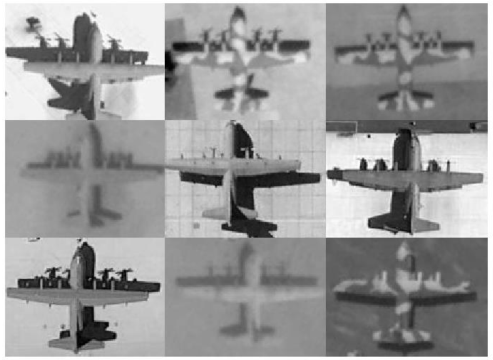
- Probabilistic fusion of features
- Weibull-calibrated SVM

[1] Bapst, A., Tran, J., Koch, M.W., Moya, M.M., Swahn, R. “Open Set Recognition of Aircraft in Aerial Imagery using Synthetic Template Models”, *Proc. SPIE*, 2017.

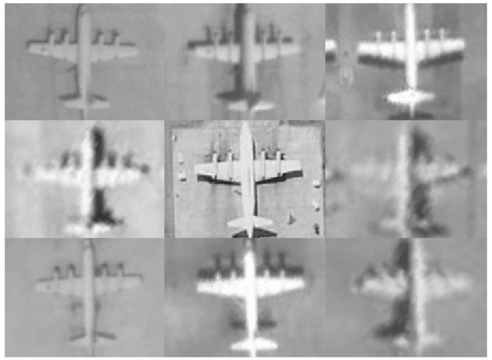
Data Set

- Google Earth images with labelled locations of various aircraft
- Reliable object detector for getting cropped images exists.

Known knowns

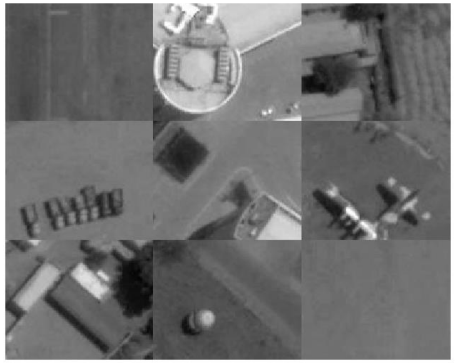


Lockheed C-130

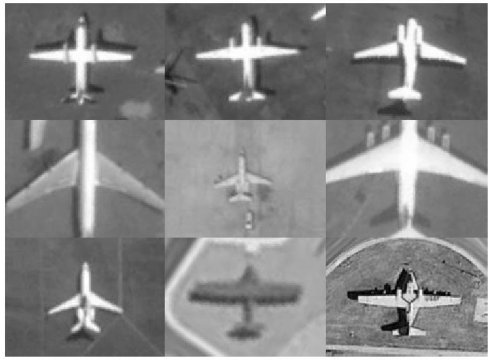


Lockheed P-3

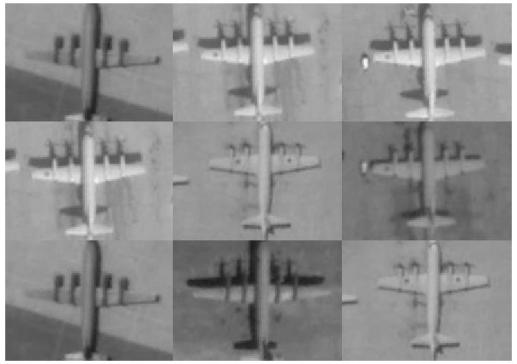
Known unknowns



Background



Confusers

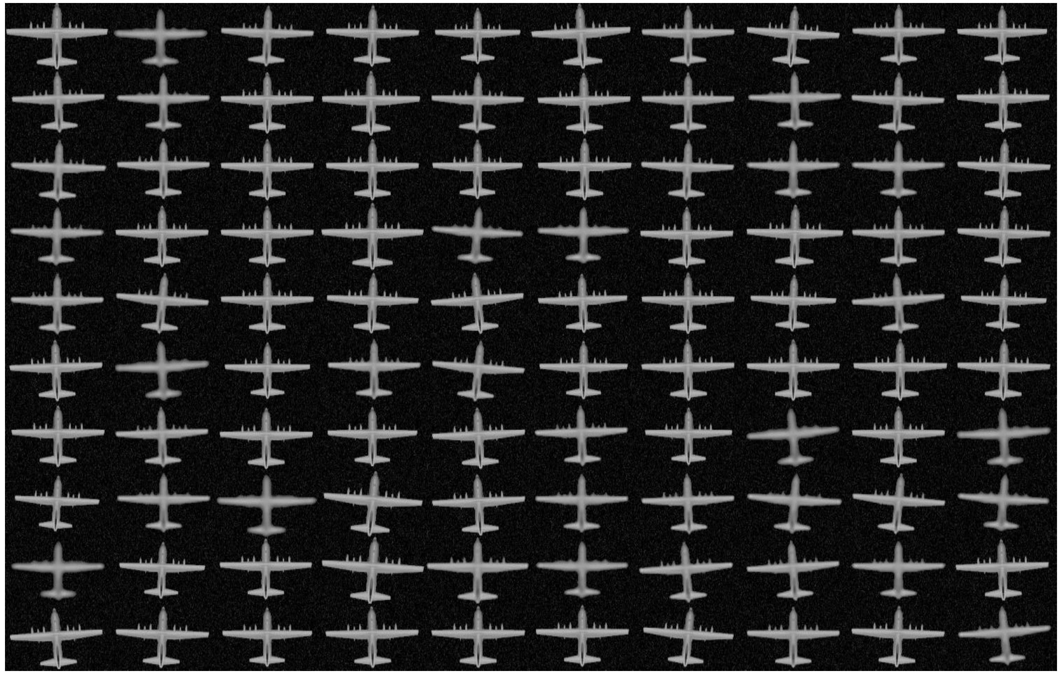


Ilyushin IL-38

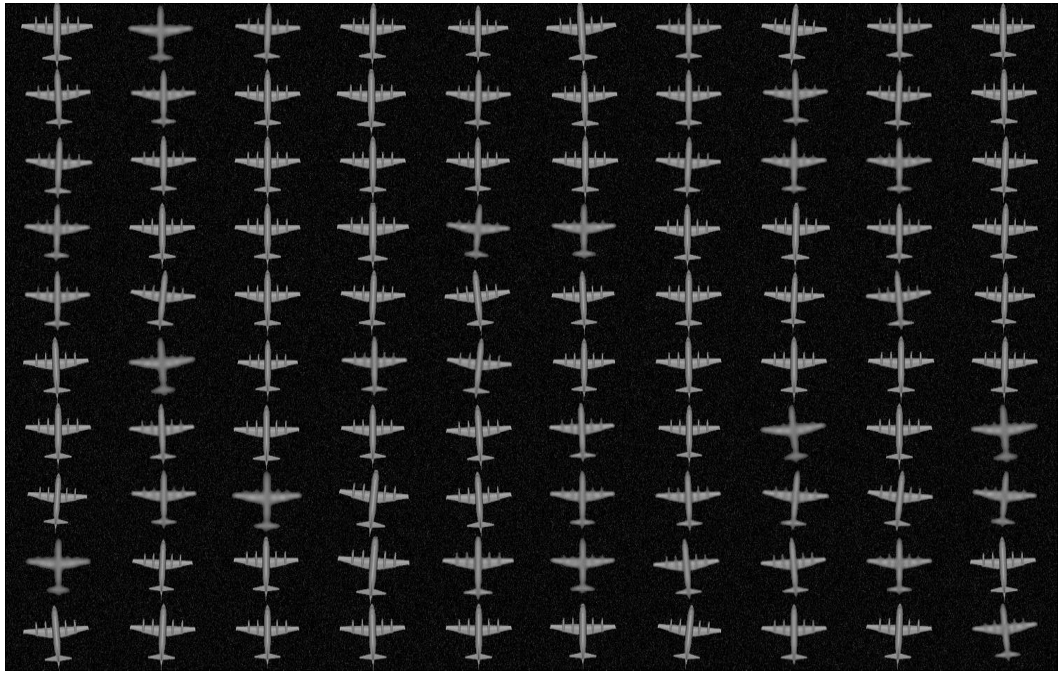
Synthetic Data

- Obtained CAD models online and generated a wide variety of data augmentations
  - Shadow angle rendering
  - Rotation/scale/resolution/translation variation
  - Addition of Gaussian noise.

Lockheed C-130



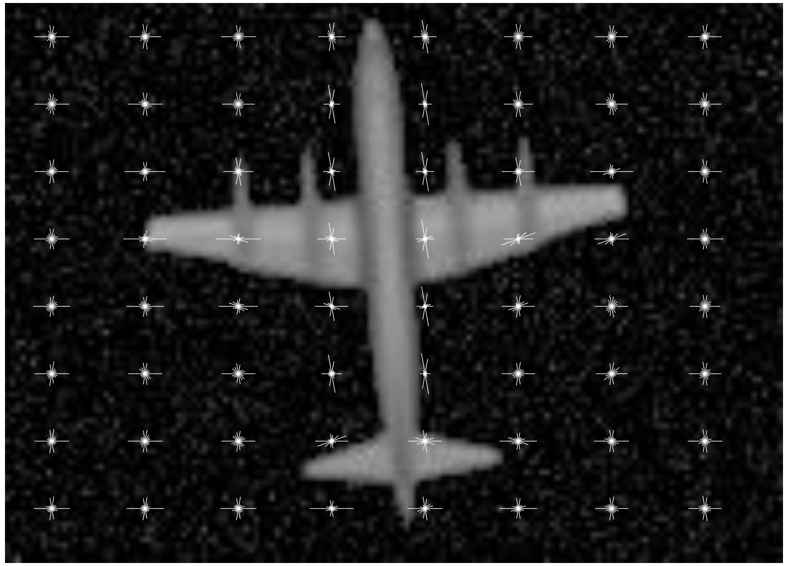
Lockheed P-3



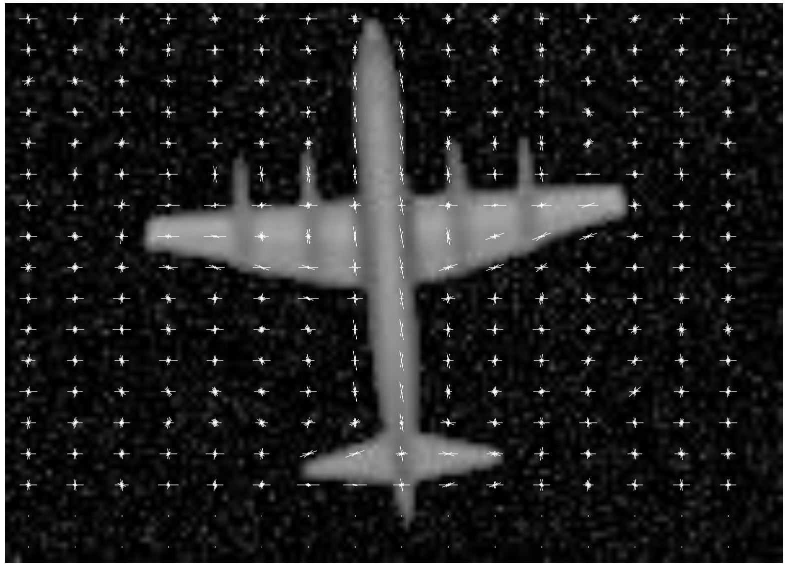
Feature Engineering

- Histogram-of-Oriented Gradients (HOG) features
  - Accurately encode shape information
  - Not rotation or illumination invariant

#features = 576



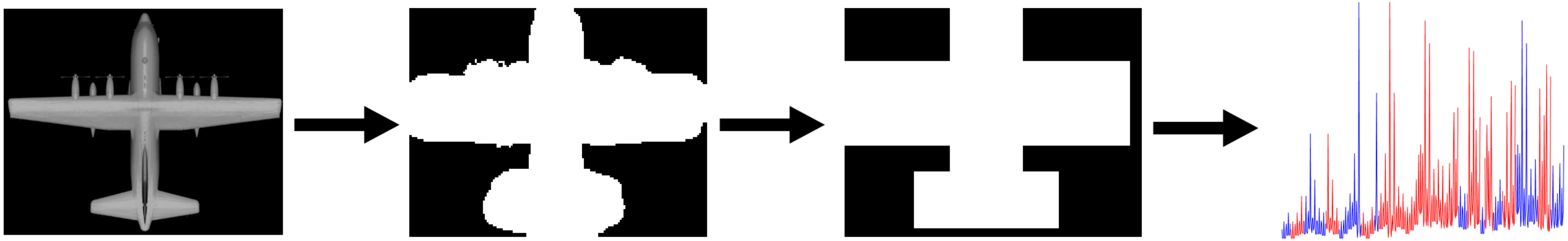
#features = 2304



Feature Engineering (cont'd)

Investigated three areas:

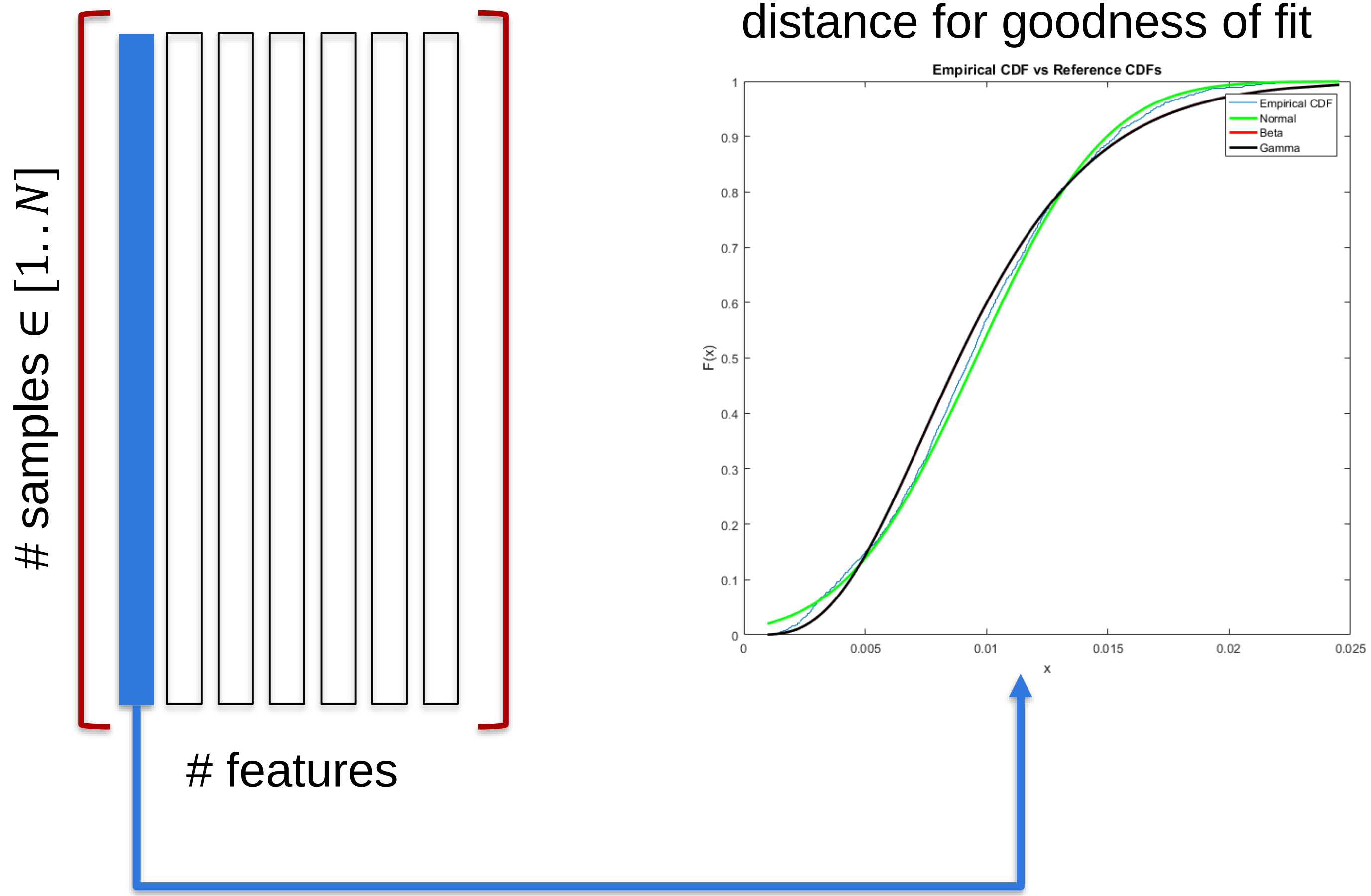
- Quantization of HOG feature values
  - Reduce noise in gradient intensity values
- PCA
  - Provides dimensionality reduction and extraction of features along axes of greatest variance
- Feature selection
  - Removal of features that lie in cells outside the object of interest



Algorithm 1: Probabilistic Fusion[1]

Target Training Data

Parametric fit to CDF using KS distance for goodness of fit



Training a PF class template

For each feature column:

1. Fit an empirical CDF

$$F_i(z_i) = Prob(Z_i \leq z_i), i \in [1 \ldots N]$$

2. Apply transformation

$$Y_i = -\log(1 - F_i(z_i))$$

3. Add up feature distributions to get fused scores

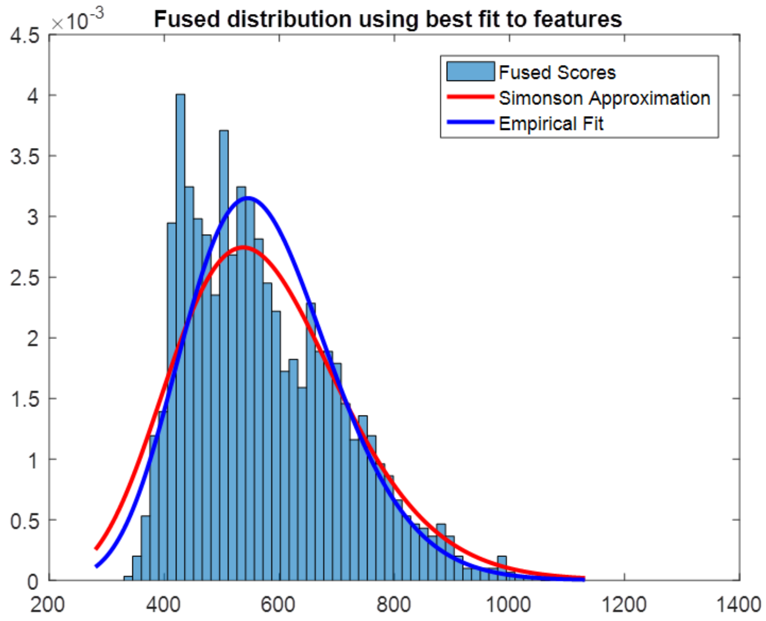
$$S_f = \sum_{i=1}^{N} Y_i$$

[1] K. M. Simonson, [Probabilistic Fusion of ATR Results] Sandia National Laboratories, (1998).

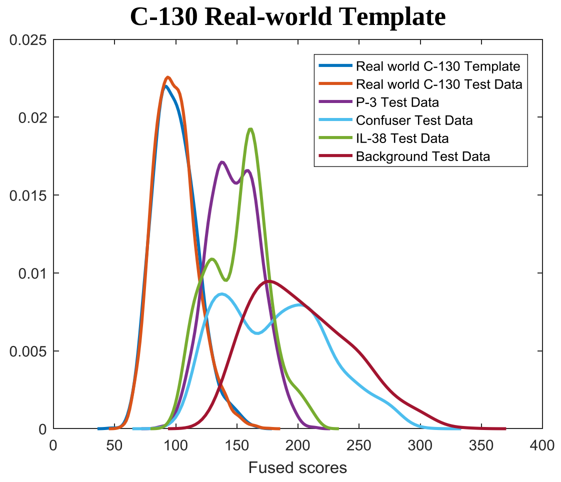
Probabilistic Fusion (cont'd)

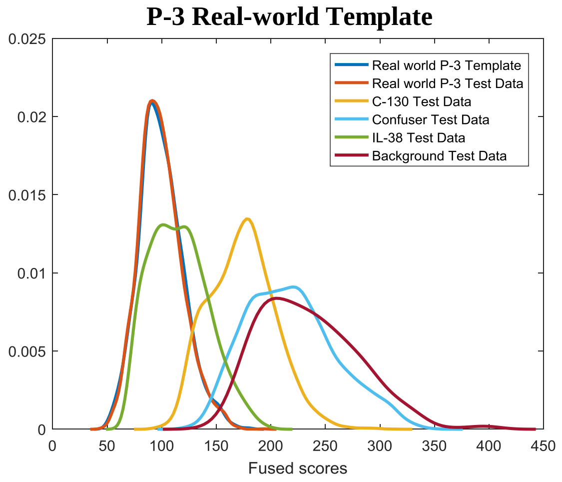
- If features are uncorrelated, the fused template scores will be gamma distributed with shape and scale parameters $r = N$and $\lambda = 1$, where N is the number of features being fused
- If features are correlated, the shape and scale parameters can be approximated as:
  - $C = \sum_{i=1}^{N} \sum_{j \neq i} \rho_{ij}$
  - $\hat{r} = \frac{N^2}{N+C}$
  - $\hat{\lambda} = \frac{N}{N+C}$

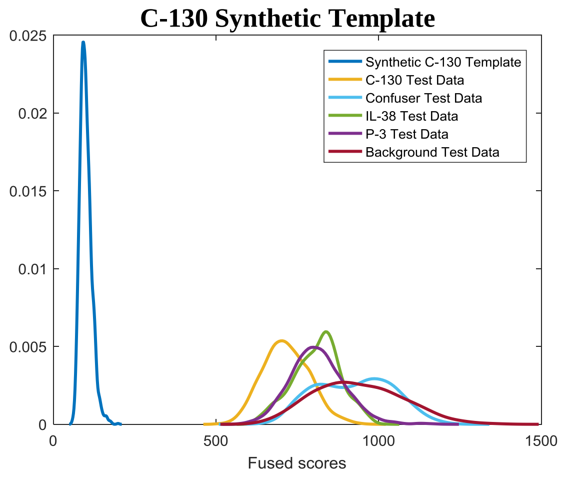
  where ρ is the correlation matrix for the target features.
- Testing the hypothesis that an unknown feature vector is a member of a given class is measured as a p-value against the template gamma distribution

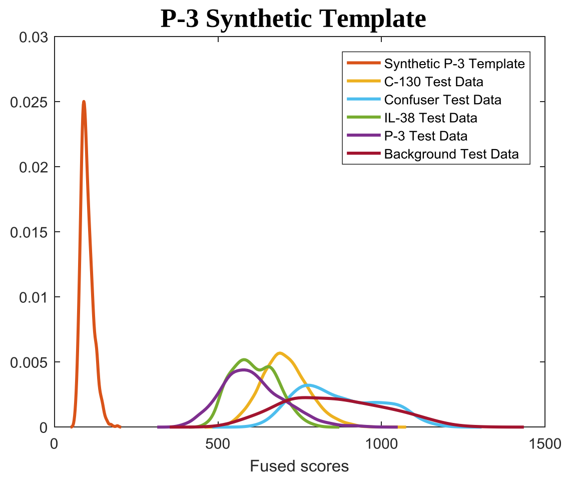


Probabilistic Fusion Results

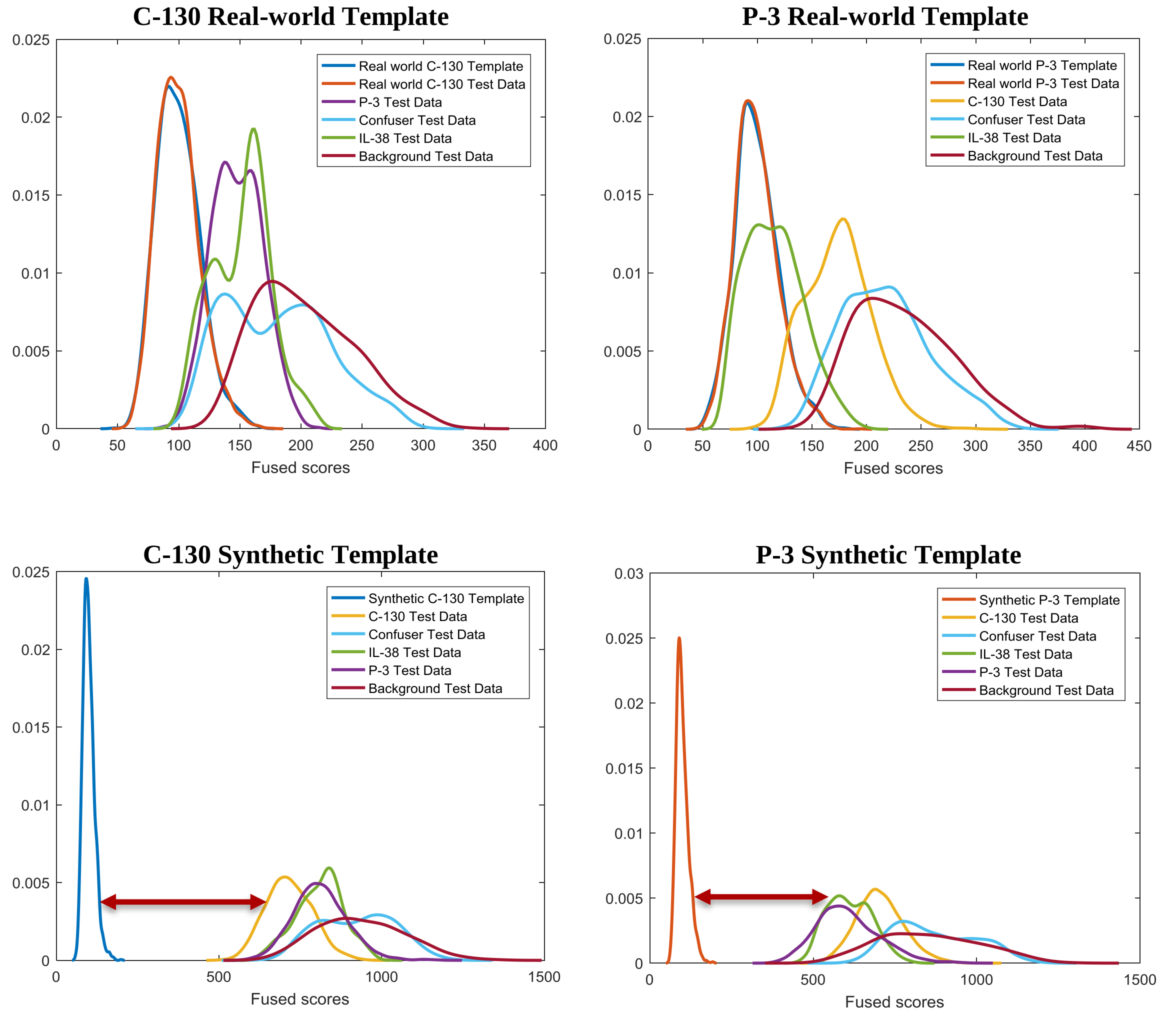






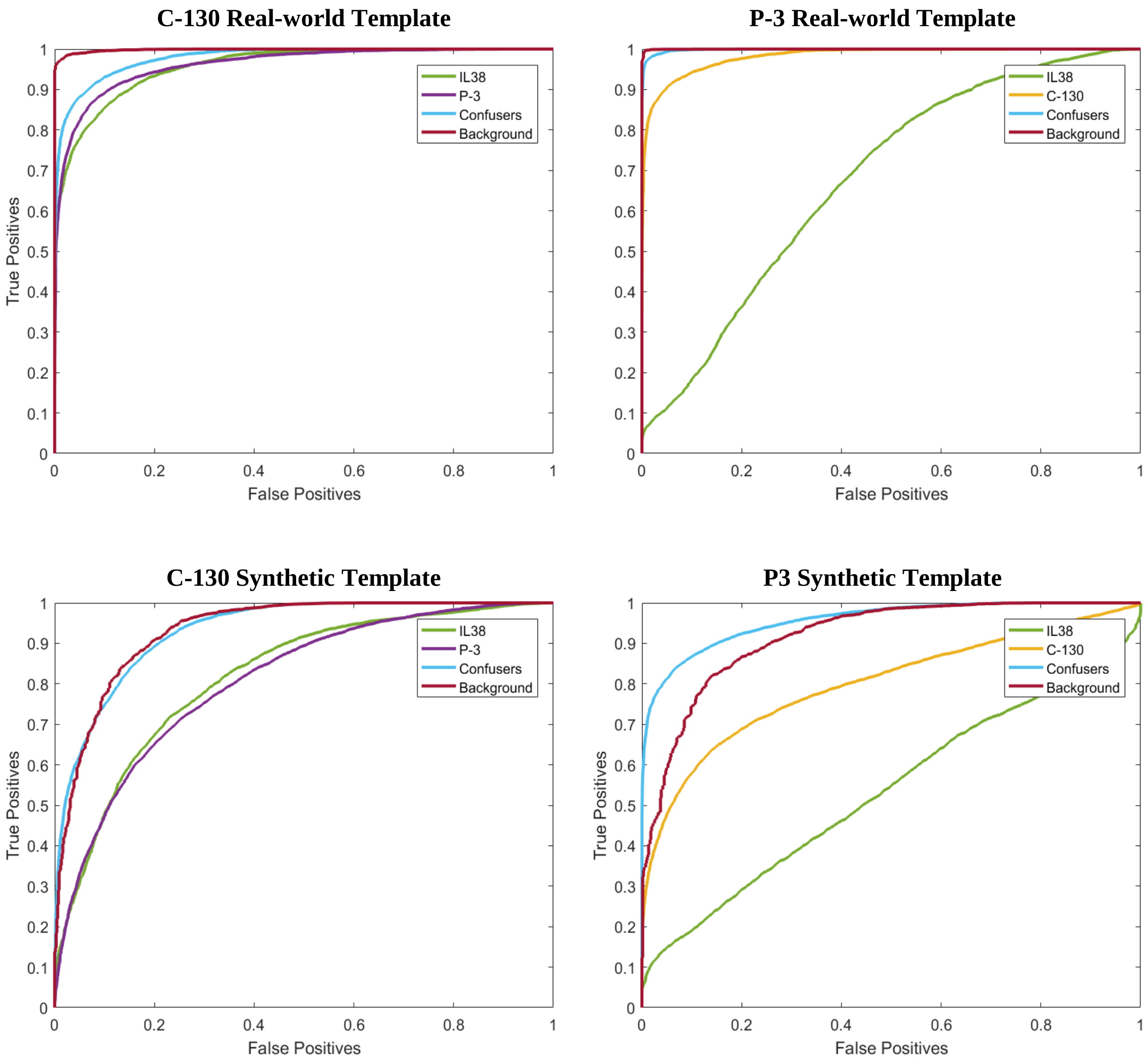


Probabilistic Fusion Results



Still work to be done with synthetic data…

Probabilistic Fusion Results

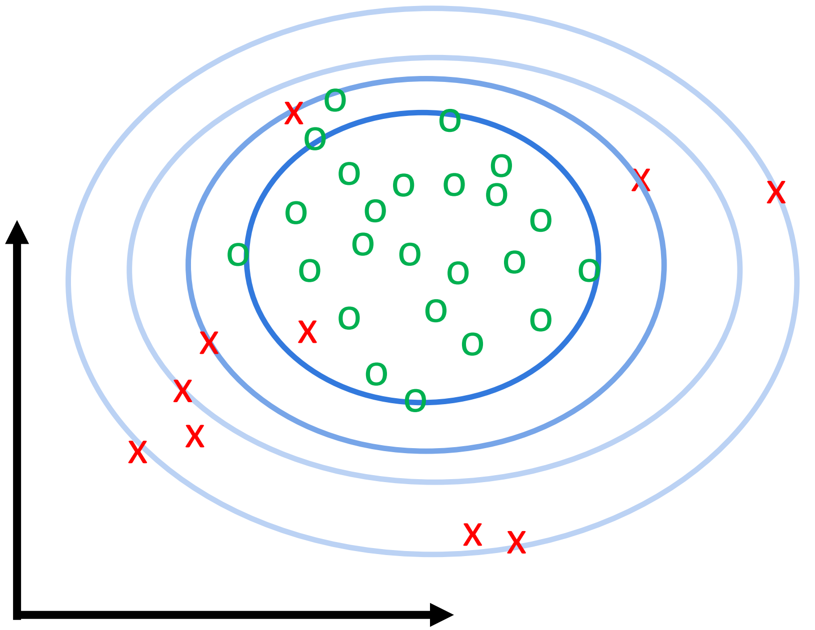


Algorithm 2: Weibull-calibrated SVM

- Combines one-class SVMs with evidence from binary SVMs using training data from *known unknowns*
- Project features into kernel space using the radial basis function
  - ν is a hyperparameter that controls one-class inclusiveness

$$k(\boldsymbol{x}, \boldsymbol{x}') = e^{-\frac{\left\|x - x'\right\|^2}{2\nu^2}}$$

- Model open space as a “compact abating probability” by fitting a Weibull distribution to extreme score values



2D representation of a one-class SVM in kernel space. SVM scores fall off with distance from the cluster center.

[1] Scheirer, W. J., Jain, L. P. and Boult, T. E. (2014). Probability Models for Open Set Recognition

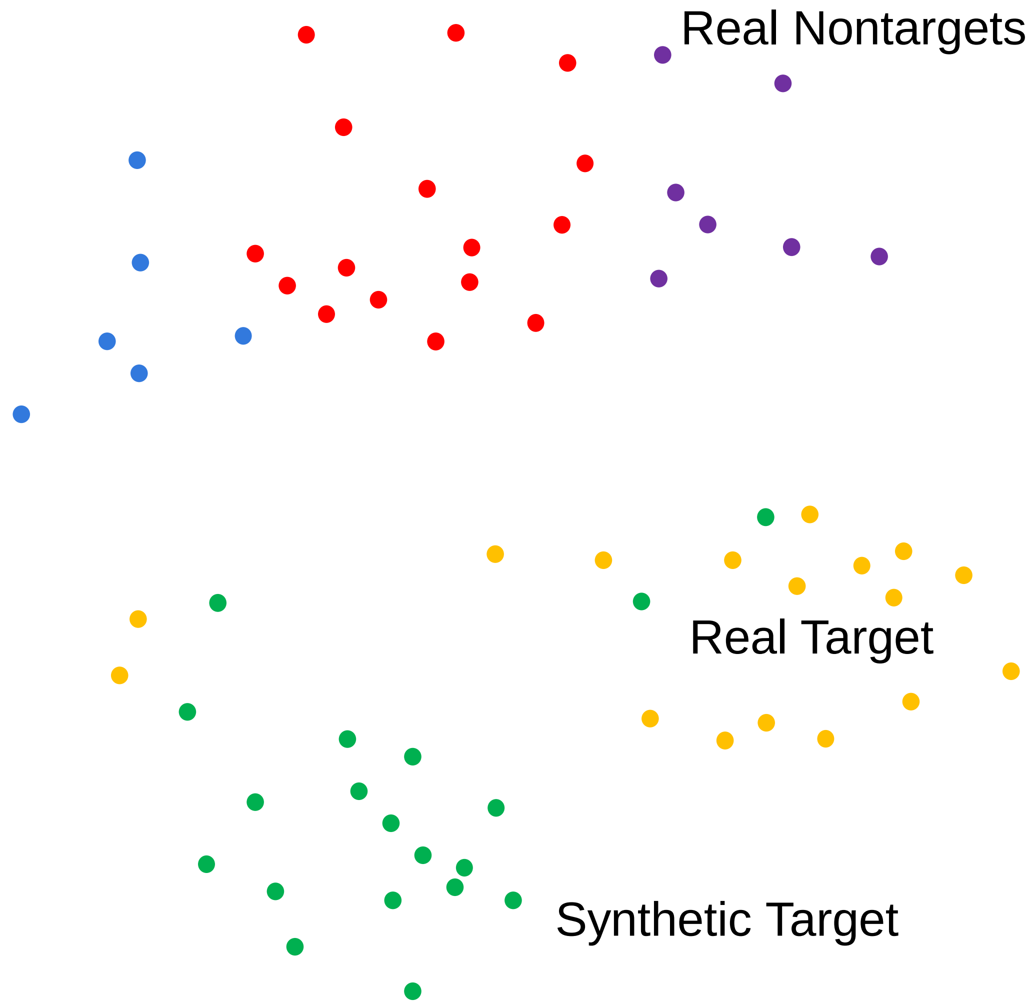
Weibull-calibrated SVM (cont'd)

Training a W-SVM model

1. Train a one-class SVM on synthetic data.
2. Fit Weibull CDF to largest one-class scores.

$$i_{match} = \begin{cases} 1 & if\ P_{match} > \sigma_t \\ 0 & else \end{cases}$$

3. Train binary SVM on "known knowns" and "known unknowns".
4. Fit Weibull CDFs to largest target scores (P_η) and reverse Weibull CDF to smallest nontarget scores (P_ψ).
5. $P_{W-SVM} = i_{match} \times P_\eta \times P_\psi$



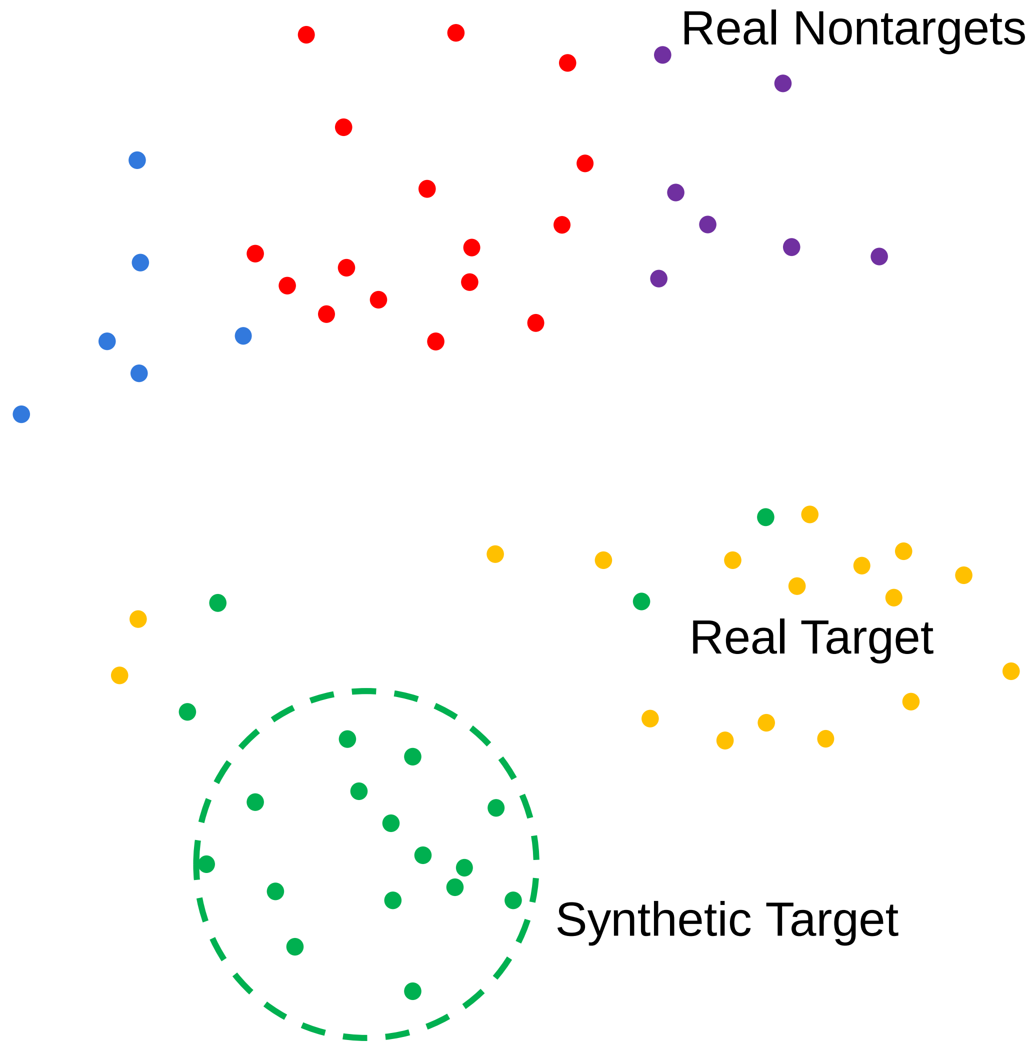
Weibull-calibrated SVM (cont'd)

Training a W-SVM model

1. Train a one-class SVM on synthetic data.
2. Fit Weibull CDF to largest one-class scores.

$$i_{match} = \begin{cases} 1 & if\ P_{match} > \sigma_t \\ 0 & else \end{cases}$$

3. Train binary SVM on "known knowns" and "known unknowns".
4. Fit Weibull CDFs to largest target scores (P_η) and reverse Weibull CDF to smallest nontarget scores (P_ψ).
5. $P_{W-SVM} = i_{match} \times P_\eta \times P_\psi$



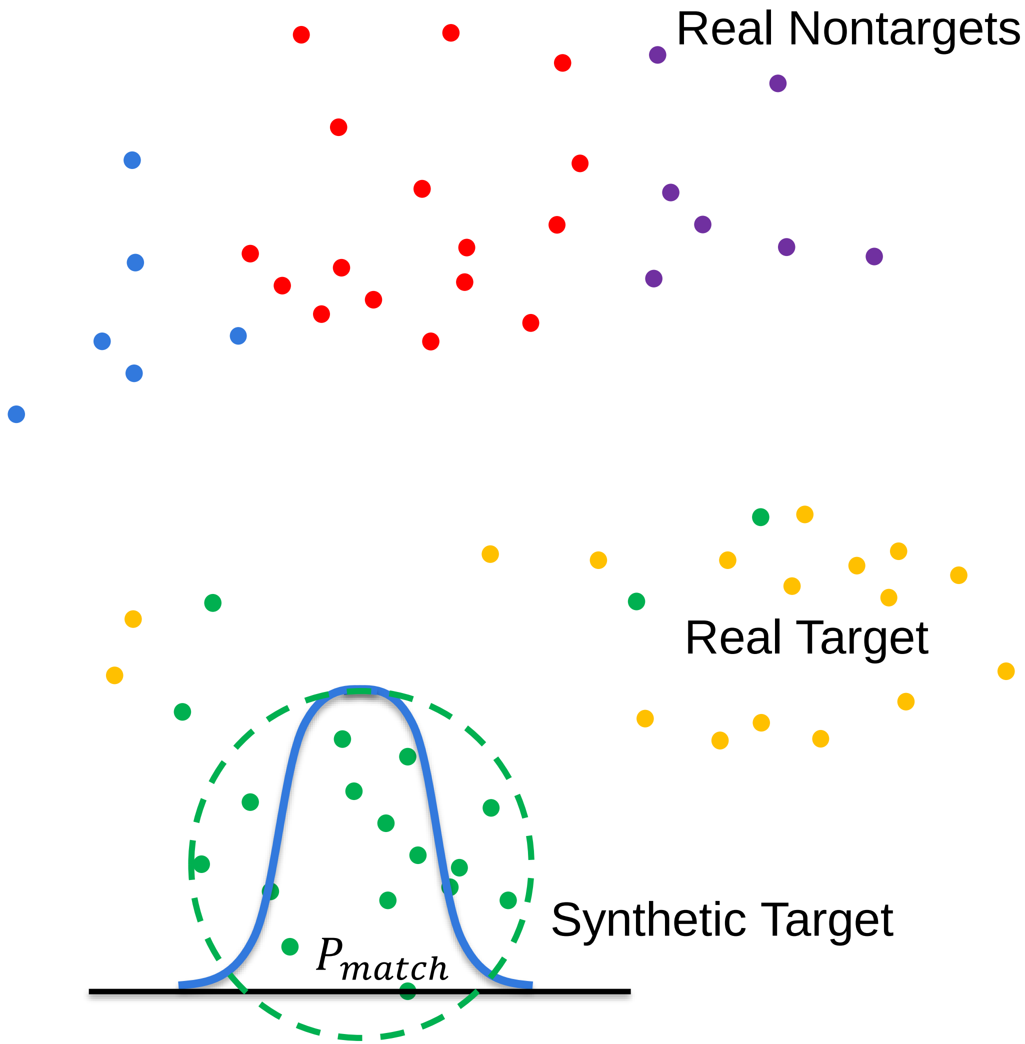
Weibull-calibrated SVM (cont'd)

Training a W-SVM model

1. Train a one-class SVM on synthetic data.
2. Fit Weibull CDF to largest one-class scores.

$$i_{match} = \begin{cases} 1 & if\ P_{match} > \sigma_t \\ 0 & else \end{cases}$$

3. Train binary SVM on “known knowns” and “known unknowns”.
4. Fit Weibull CDFs to largest target scores (P_η) and reverse Weibull CDF to smallest nontarget scores (P_ψ).
5. $P_{W-SVM} = i_{match} \times P_\eta \times P_\psi$



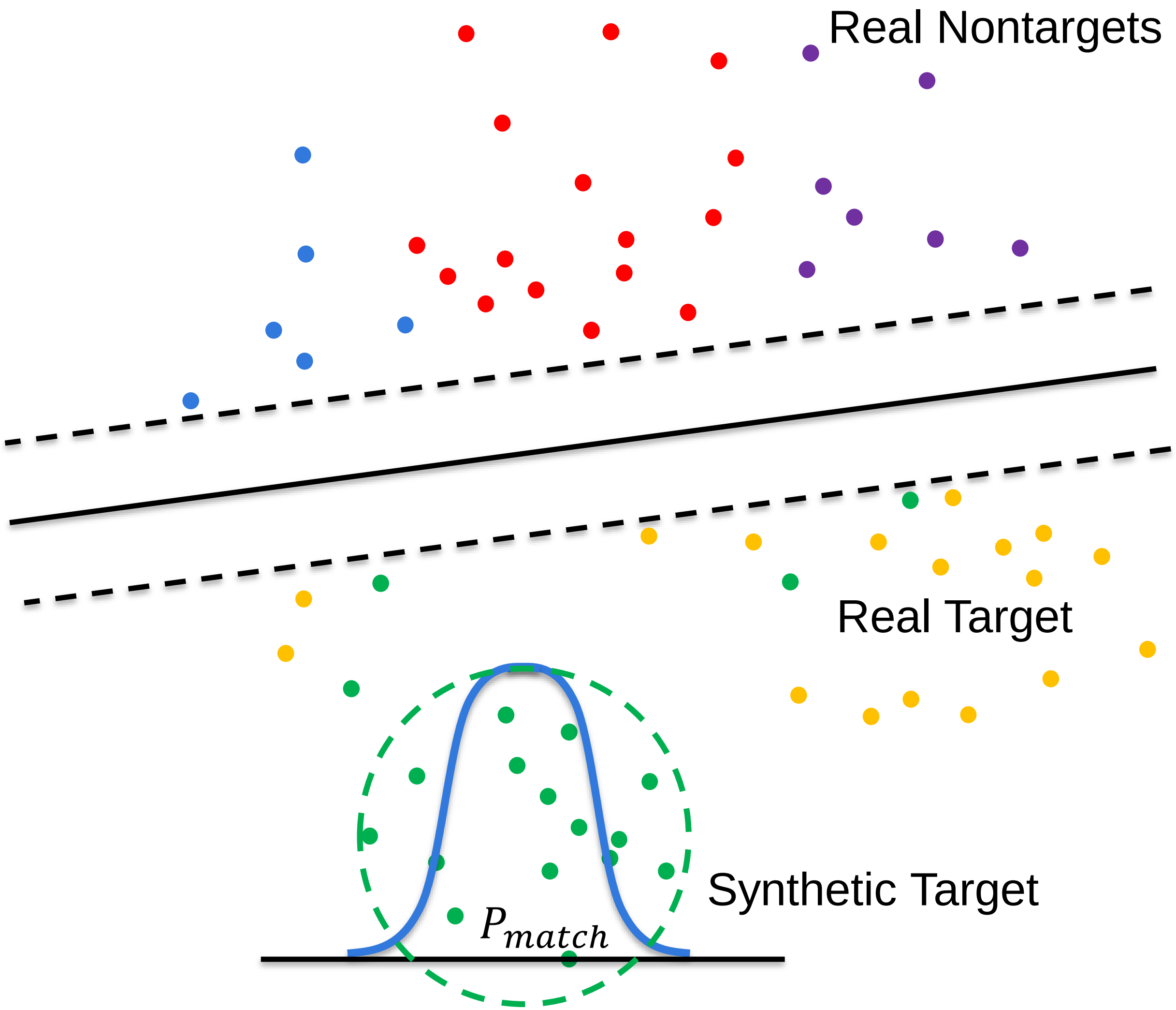
Weibull-calibrated SVM (cont'd)

Training a W-SVM model

1. Train a one-class SVM on synthetic data.
2. Fit Weibull CDF to largest one-class scores.

$$i_{match} = \begin{cases} 1 & if\ P_{match} > \sigma_t \\ 0 & else \end{cases}$$

3. Train binary SVM on "known knowns" and "known unknowns".
4. Fit Weibull CDFs to largest target scores (P_η) and reverse Weibull CDF to smallest nontarget scores (P_ψ).
5. $P_{W-SVM} = i_{match} \times P_\eta \times P_\psi$



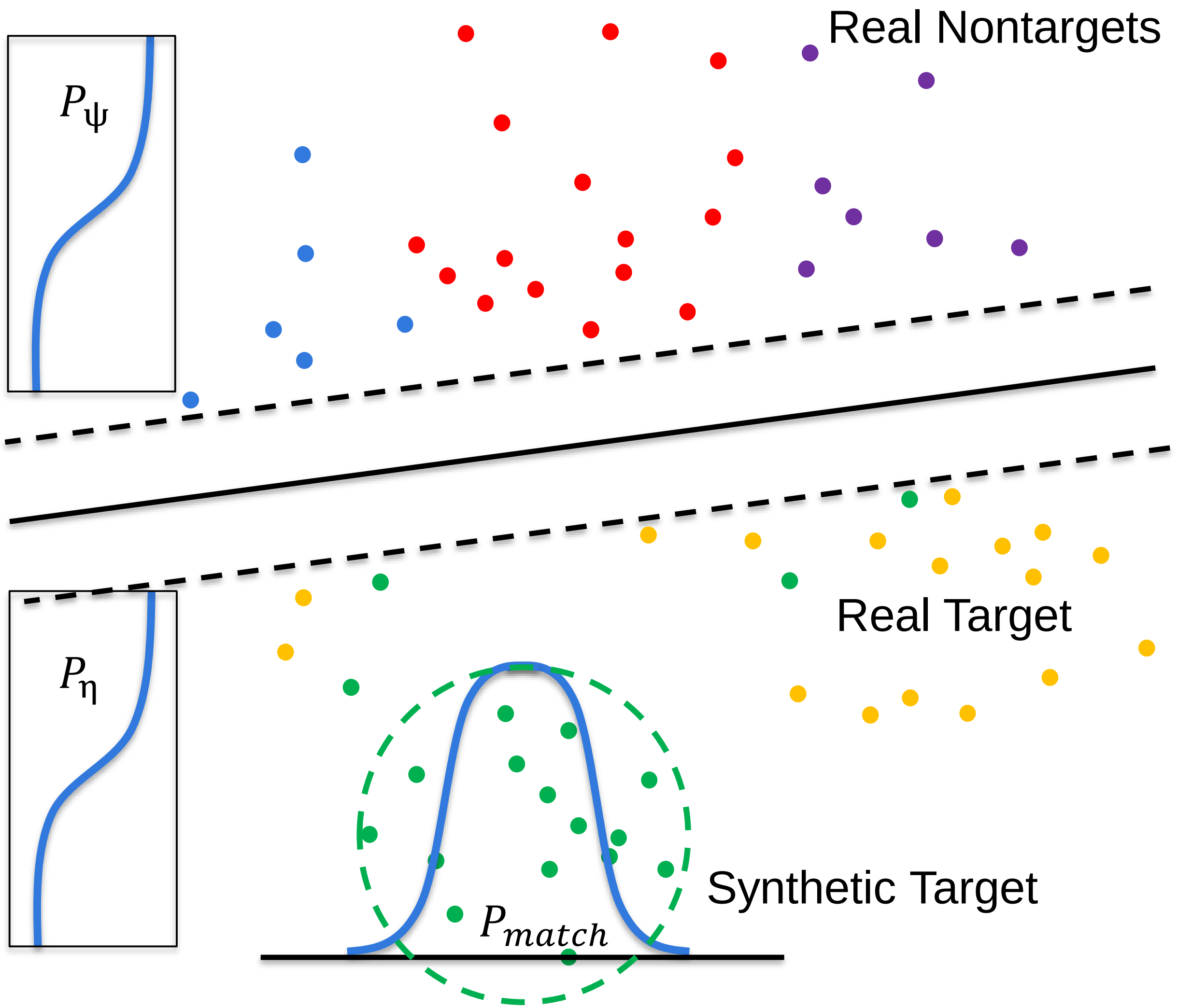
Weibull-calibrated SVM (cont'd)

Training a W-SVM model

1. Train a one-class SVM on synthetic data.
2. Fit Weibull CDF to largest one-class scores.

$$i_{match} = \begin{cases} 1 & if\ P_{match} > \sigma_t \\ 0 & else \end{cases}$$

3. Train binary SVM on "known knowns" and "known unknowns".
4. Fit Weibull CDFs to largest target scores (P_η) and reverse Weibull CDF to smallest nontarget scores (P_ψ).
5. $P_{W-SVM} = i_{match} \times P_\eta \times P_\psi$



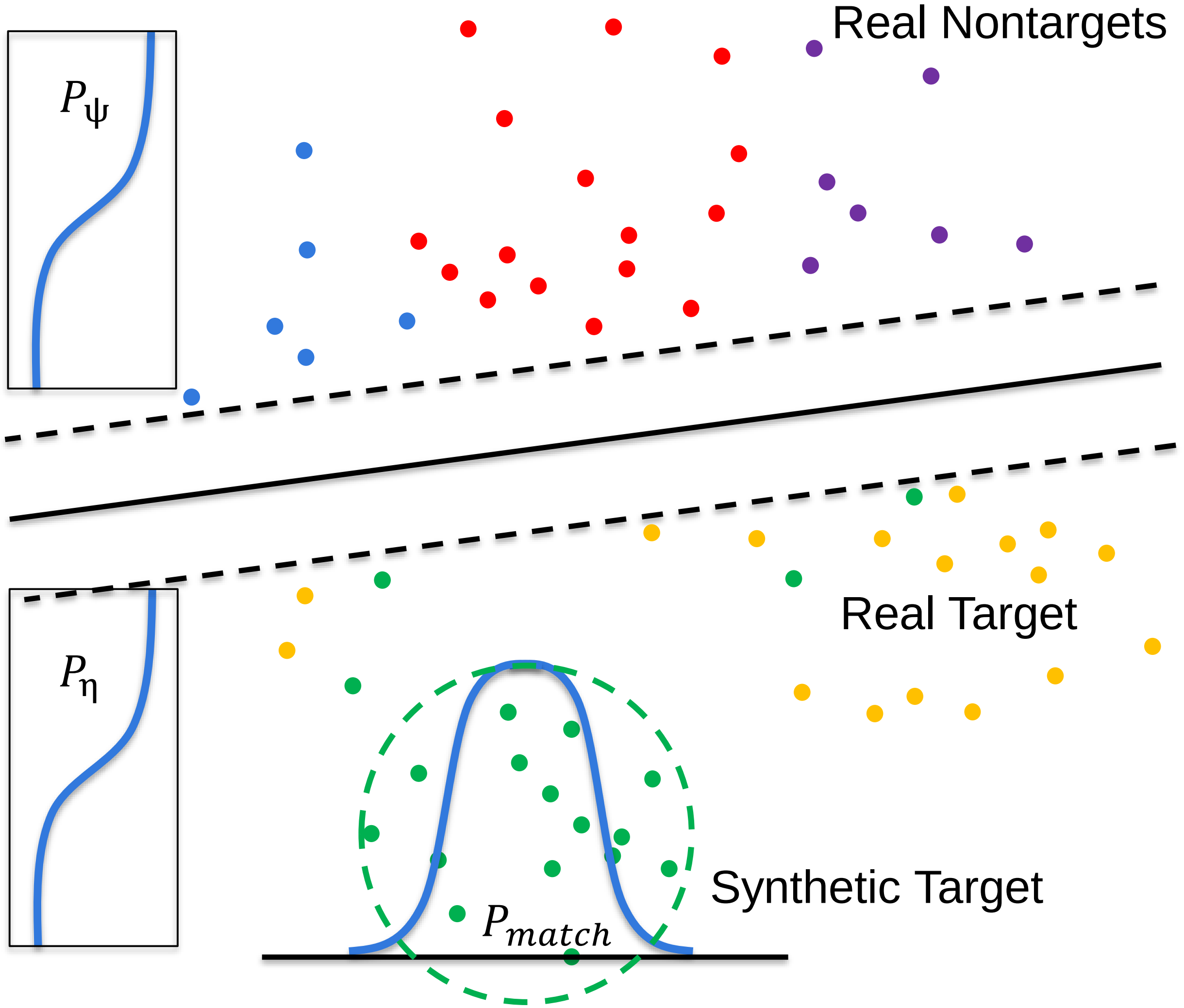
Weibull-calibrated SVM (cont'd)

Training a W-SVM model

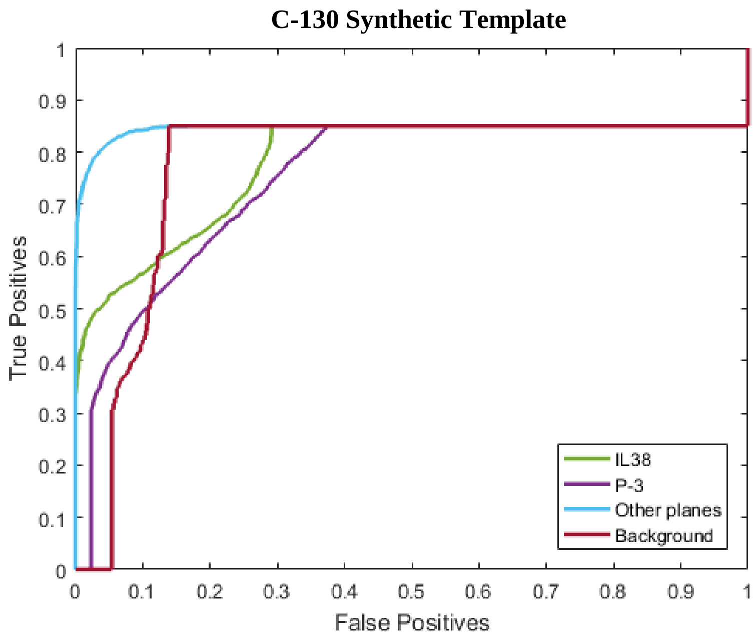
1. Train a one-class SVM on synthetic data.
2. Fit Weibull CDF to largest one-class scores.

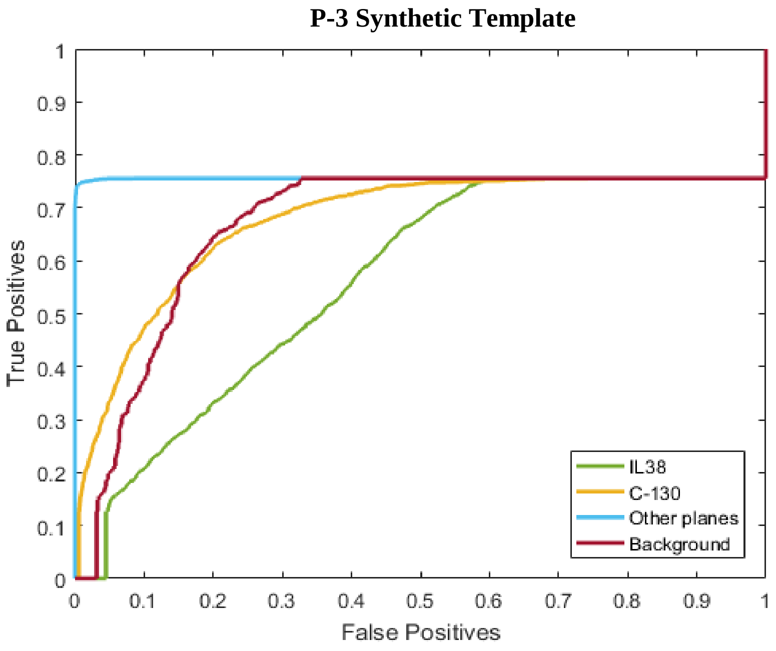
$$i_{match} = \begin{cases} 1 & if\ P_{match} > \sigma_t \\ 0 & else \end{cases}$$

3. Train binary SVM on "known knowns" and "known unknowns".
4. Fit Weibull CDFs to largest target scores (P_η) and reverse Weibull CDF to smallest nontarget scores (P_ψ).
5. $P_{W-SVM} = i_{match} \times P_\eta \times P_\psi$



Weibull-calibrated SVM results





- Thanks!