Hemi-Supervised Detection

Don Hush

ISR-2, LANL

Detection

Detection Problems are Ubiquitous: threats, disease, fraud, anomalies, structural failure, proliferation, intrusions, military targets, IEDs, bio-markers, ...

Detection

Detection Problems are Ubiquitous: threats, disease, fraud, anomalies, structural failure, proliferation, intrusions, military targets, IEDs, bio-markers, ...

- Detection Problems = Binary Classification Problems where
  - Class 1 = “target” = events we want to detect
  - Class 0 = “background/clutter” = everything else

Detection

Detection Problems are Ubiquitous: threats, disease, fraud, anomalies, structural failure, proliferation, intrusions, military targets, IEDs, bio-markers, ...

- Detection Problems = Binary Classification Problems where
  - Class 1 = "target" = events we want to detect
  - Class 0 = "background/clutter" = everything else

- Task: build a detector

$$f : X \to \{0, 1\}$$

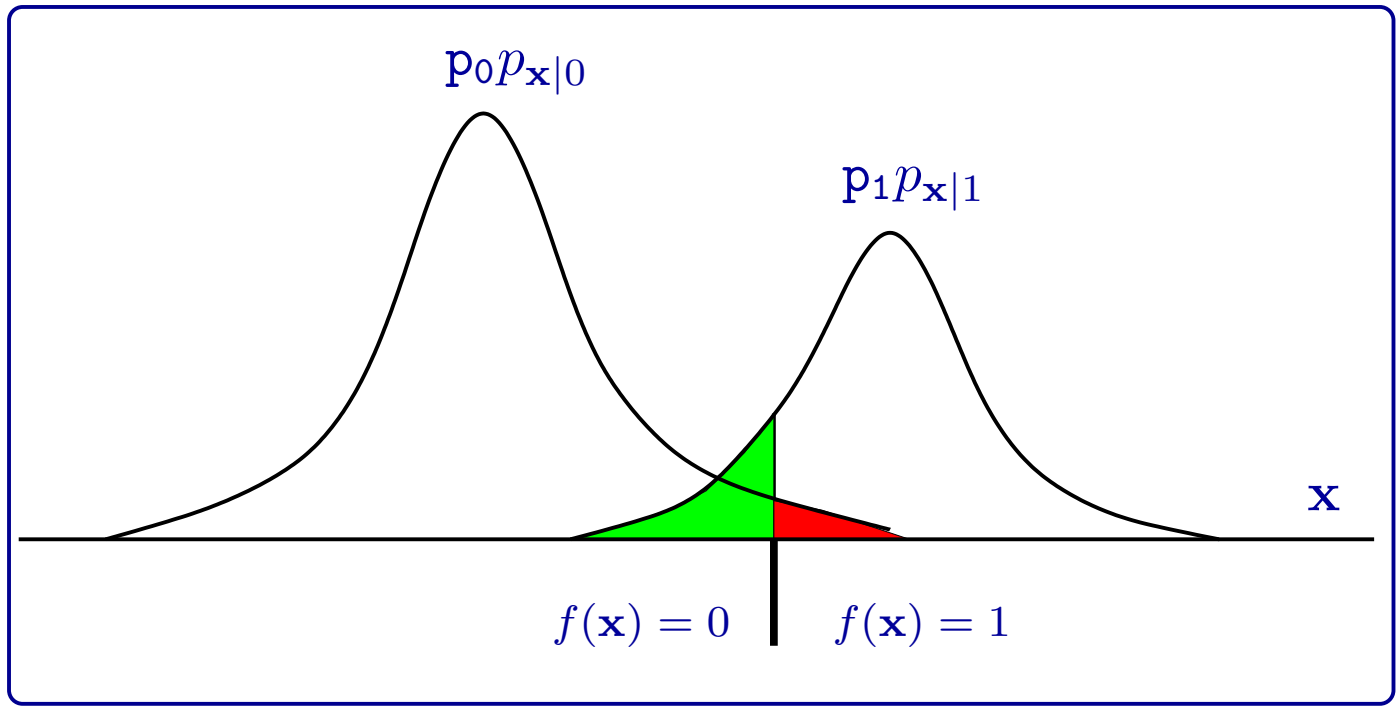
Often $\mathbf{x}$ is a d-dimensional vector, i.e. $\mathbf{x} = (x_1, x_2, ..., x_d)$, $x_i \in \mathbb{R}$.

Design Goal: small error rate

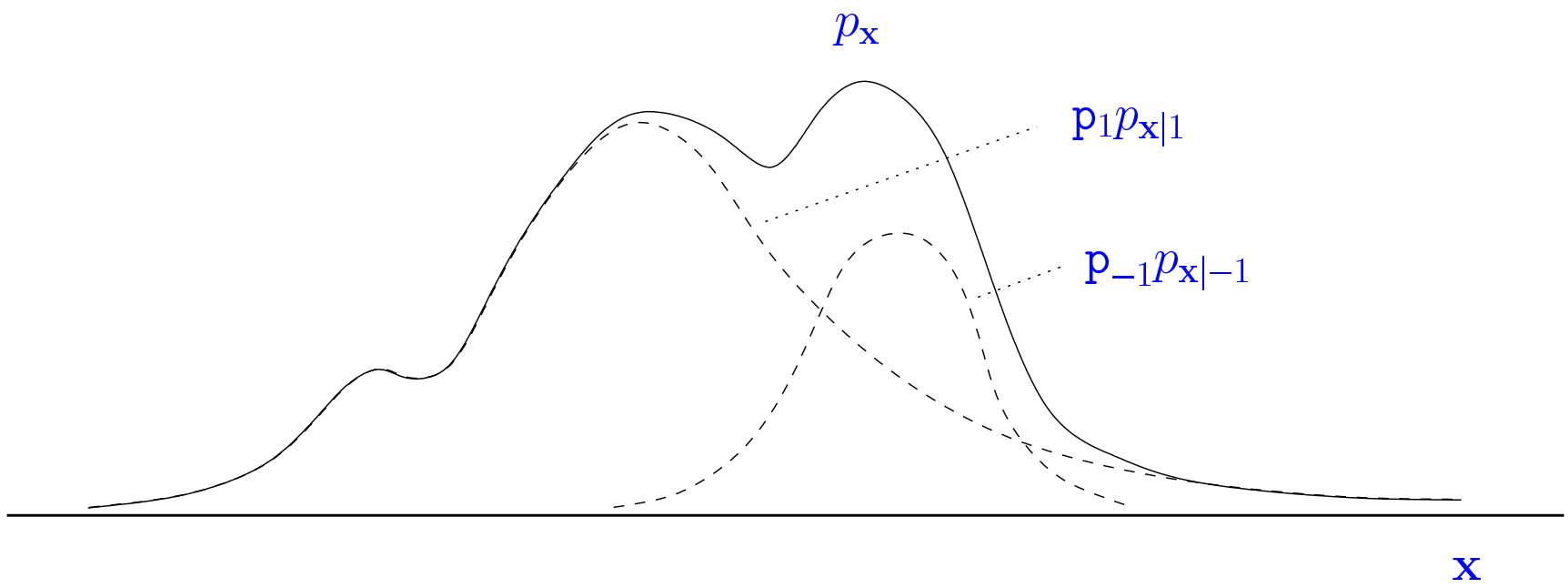
Assume $(\mathbf{x}, y)$ is a r.v. distributed according to probability density p

$$e(f) := E_p[I(f(\mathbf{x}) \neq y)] = \underbrace{\mathtt{p}_1 \int_{f(\mathbf{x})=0} p_{\mathbf{x}|1}(\mathbf{x})d\mathbf{x}}_{missed\ detection\ rate} + \underbrace{\mathtt{p}_0 \int_{f(\mathbf{x})=1} p_{\mathbf{x}|0}(\mathbf{x})d\mathbf{x}}_{false\ alarm\ rate}$$

$$= \mathtt{p}_1 e_1(f) + \mathtt{p}_0 e_0(f)$$



$p_{\mathbf{x}} = \mathtt{p}_1 p_{\mathbf{x}|1} + \mathtt{p}_0 p_{\mathbf{x}|0}$ = **x**-density = density for unlabeled observations

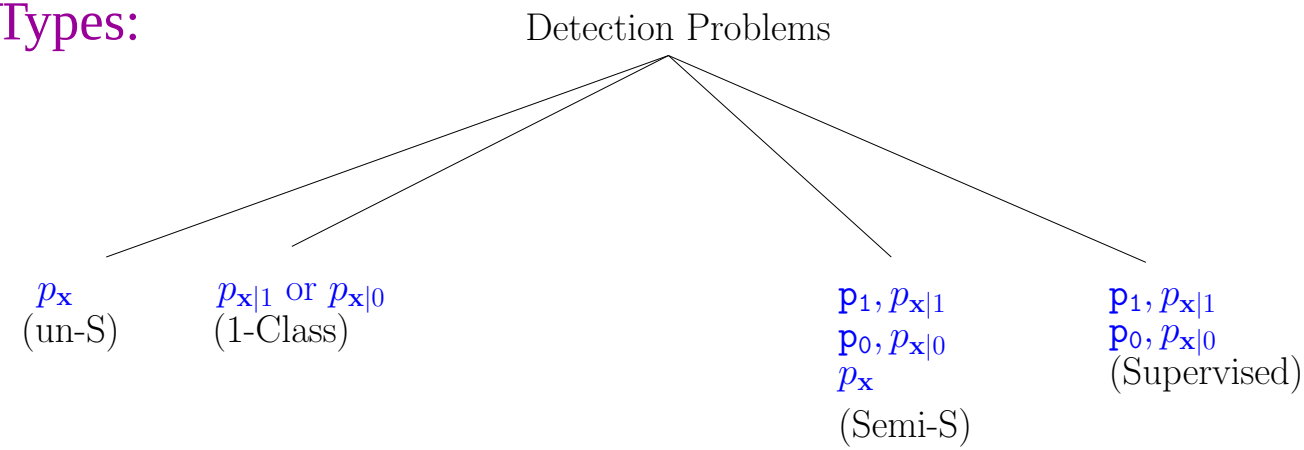


Varieties of Detection Problems

Information Sources:

- first principles
- empirical

Information Types:

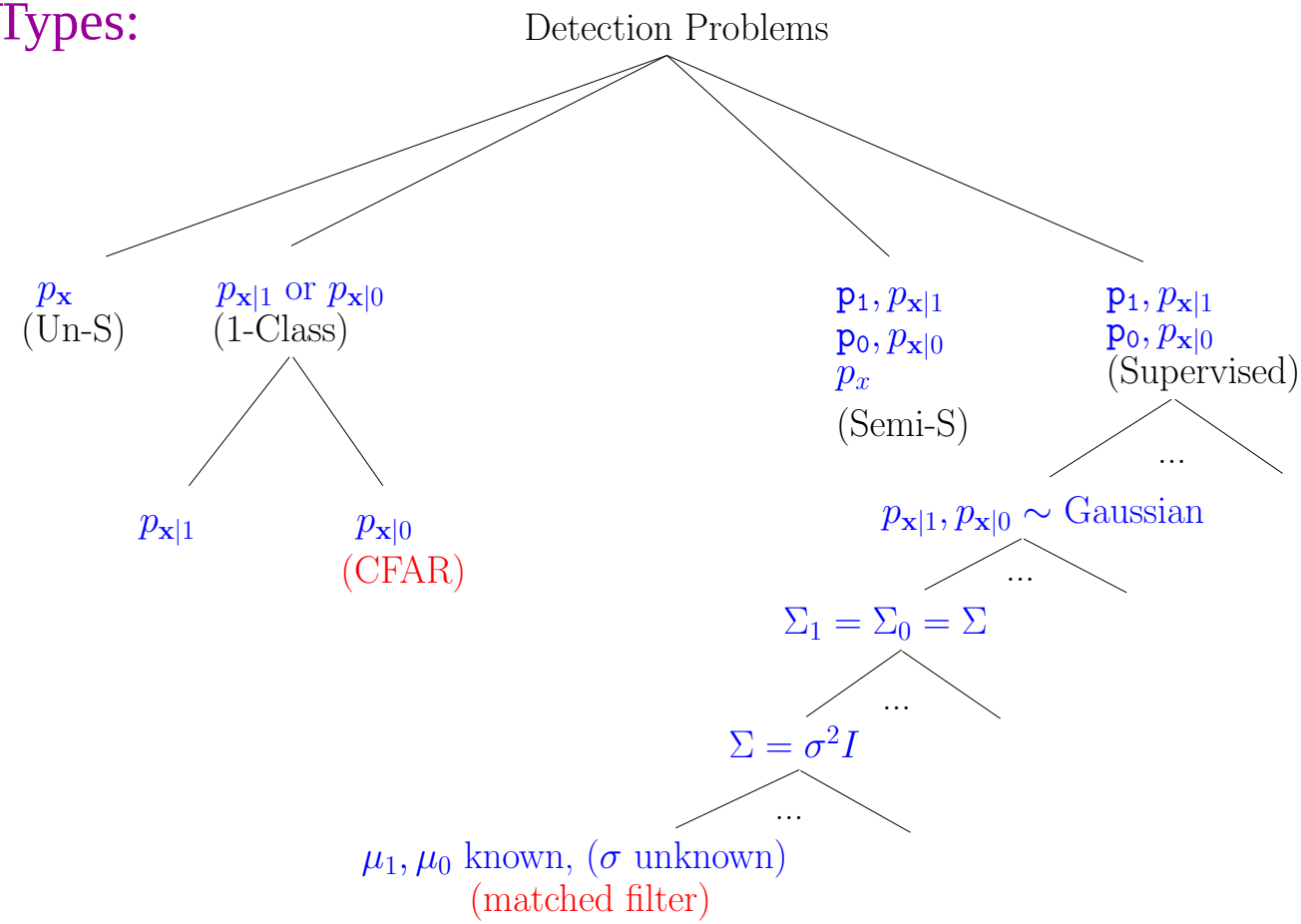


Varieties of Detection Problems

Information Sources:

- first principles
- empirical

Information Types:



Lopsided Detection

- target class: significant prior information available about $p_{\mathbf{x}|1}$ (e.g. sample data)
- background class: little or no prior information about $p_{\mathbf{x}|0}$ because ...
  - deployed environment not known ahead of time
  - different background for different deployments
  - background may change with time

Examples:

- target detection in remote sensing (e.g. images, sensor networks, ...)
- cancer detection (and most other medical detection tasks)
- insider threat detection
- category detection in text documents
- behavior detection in computer network traffic
- land cover type detection in mult-spectral images
- preferred web page detection for individual users
- near-failure detection for physical structures
- (marketing) potential future customer detection based on current customer database
- (insurance) detecting “at risk” customers

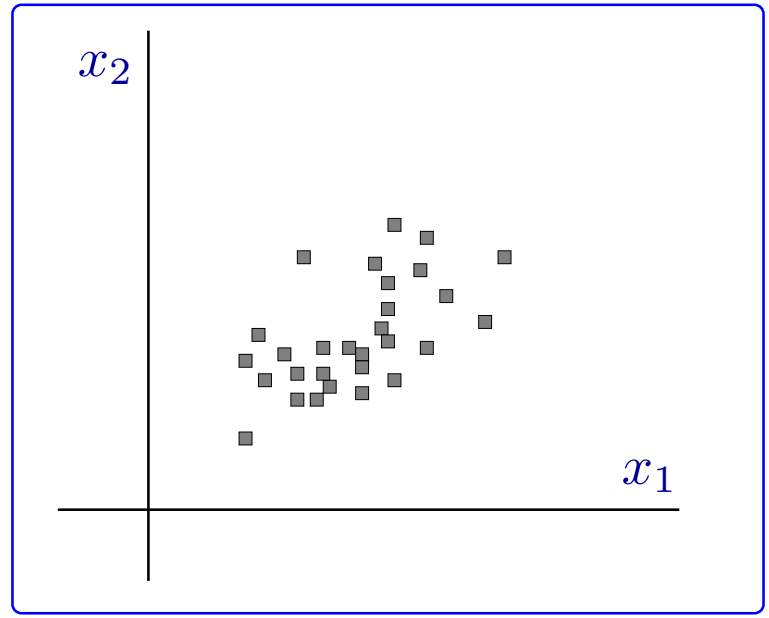
Common Approach: 1-Class Design Methods

1-Class Design Methods

- Objective: 1-class control (either e_1 or e_0)

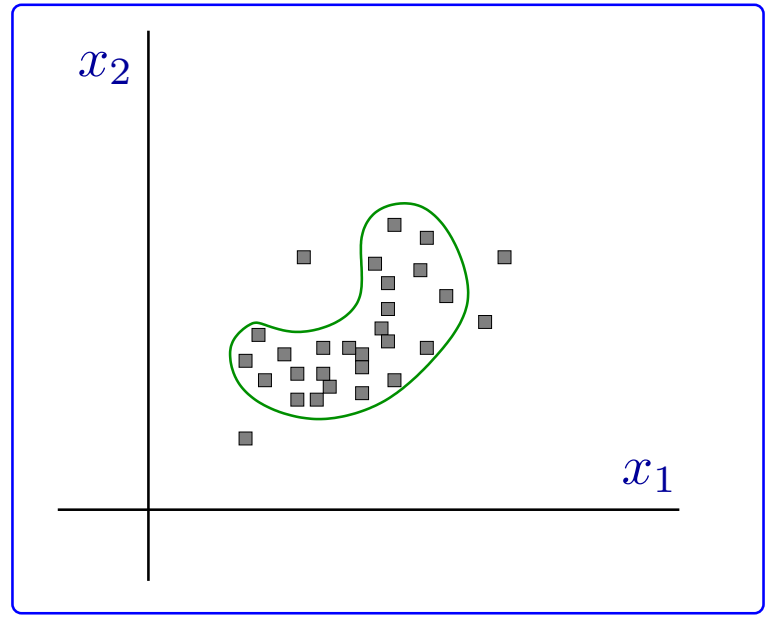
1-Class Design Methods

- Objective: 1-class control (either e_1 or e_0)



1-Class Design Methods

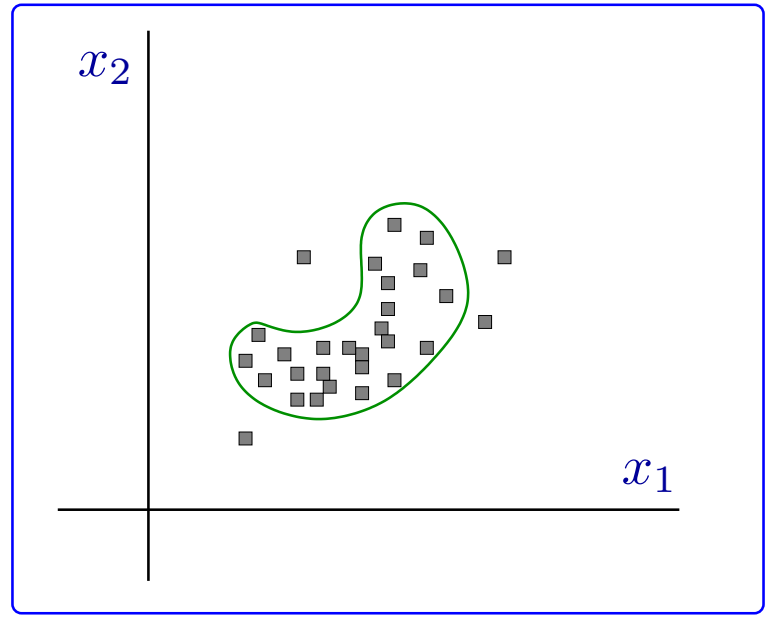
- Objective: 1-class control (either e_1 or e_0)



  - Design f to control one of the class error rates (e.g. missed detections)
  - *and* minimize the volume of the set $\{\mathbf{x} : f(\mathbf{x}) = 1\}$.

1-Class Design Methods

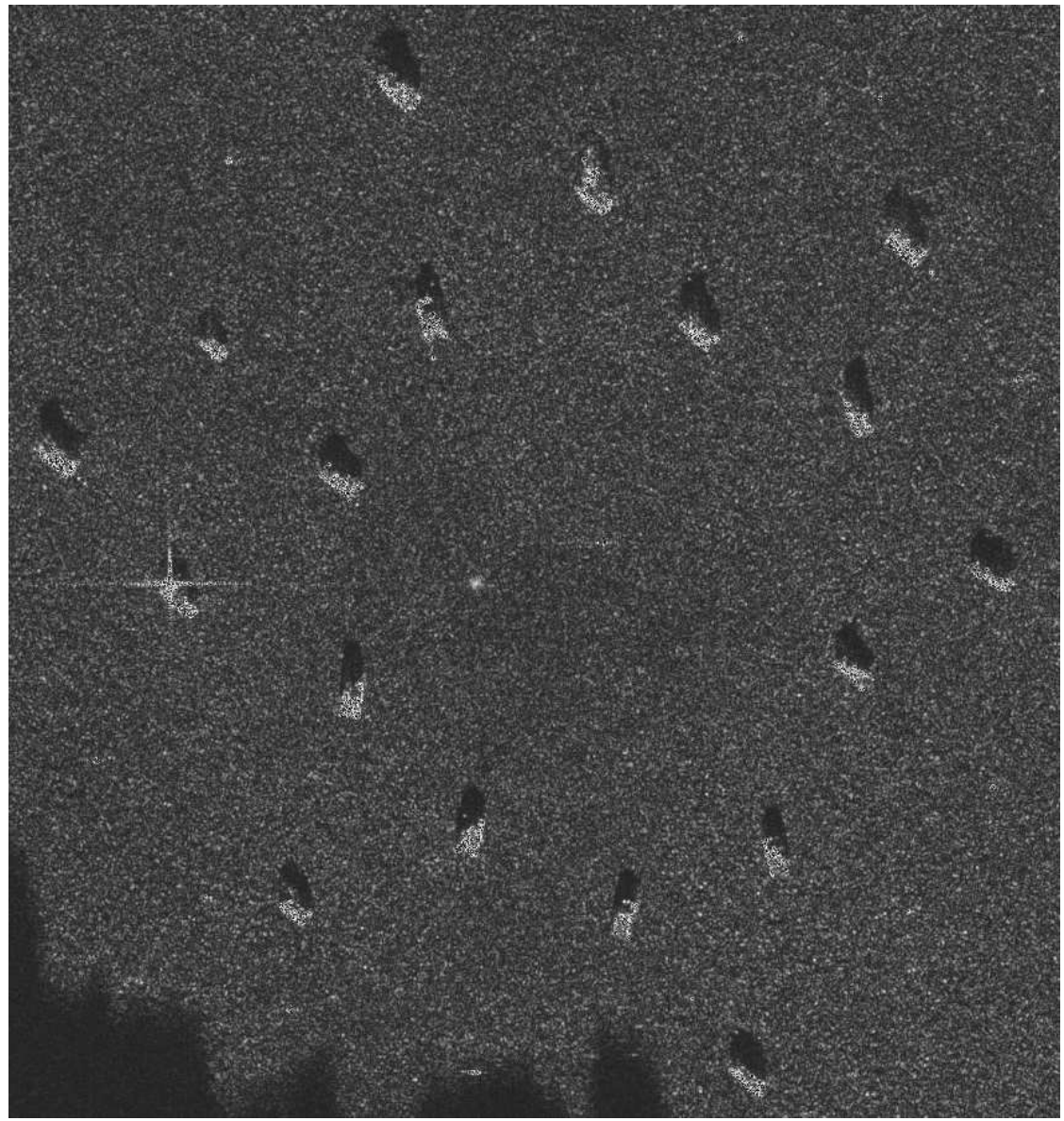
- Objective: 1-class control (either e_1 or e_0)

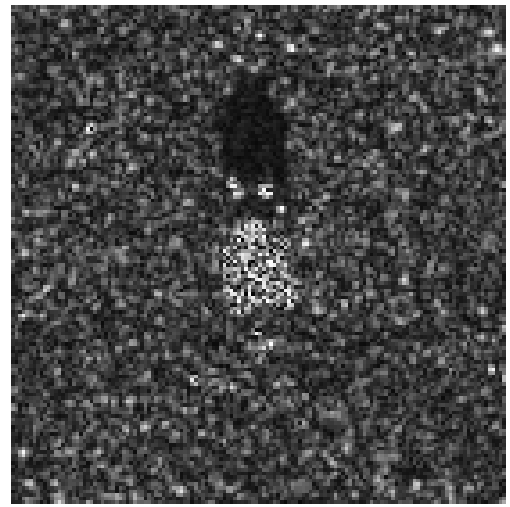


  - Design f to control one of the class error rates (e.g. missed detections)
  - *and* minimize the volume of the set $\{\mathbf{x} : f(\mathbf{x}) = 1\}$.

- Solution Methods: density estimation (+ threshold), clustering (+ thresholds), template matching, 1-class SVM, DLD-SVM, CFAR, ...

SAR Image Segmentation



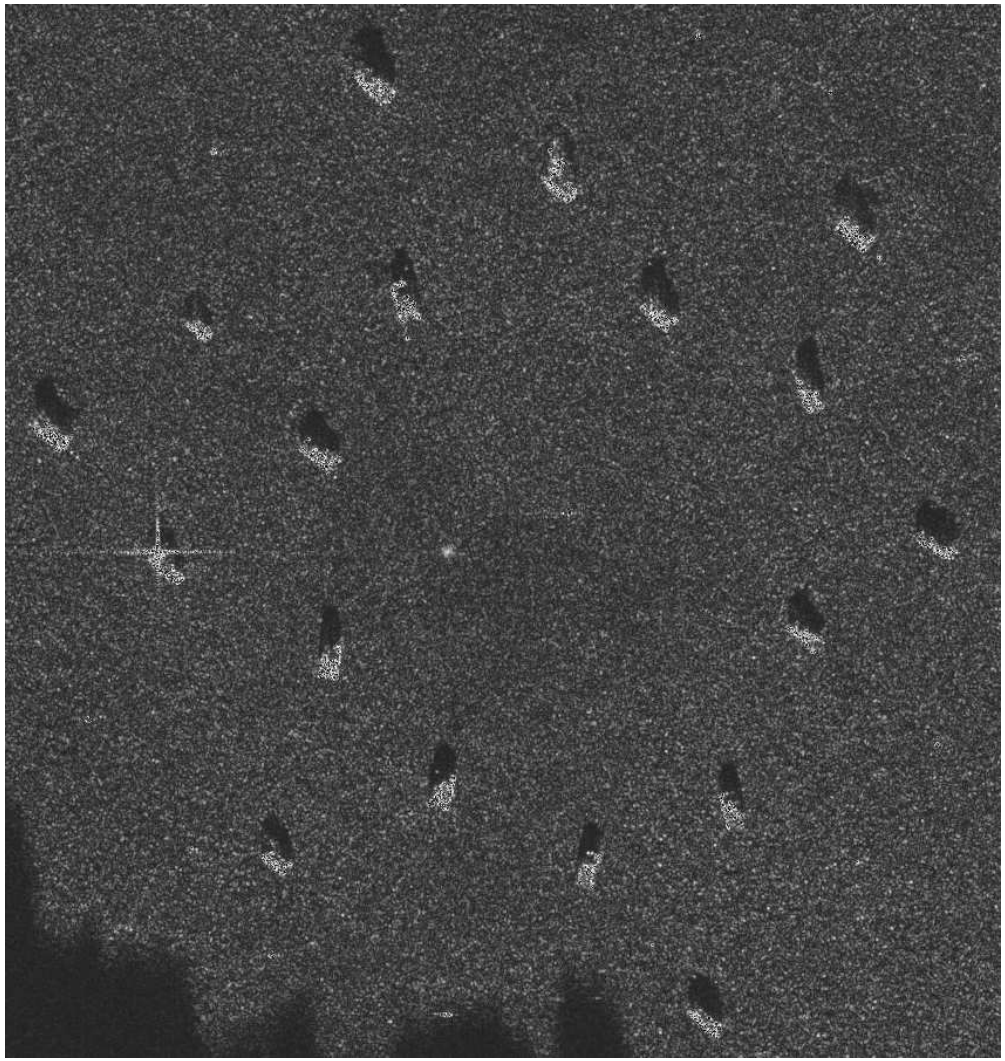


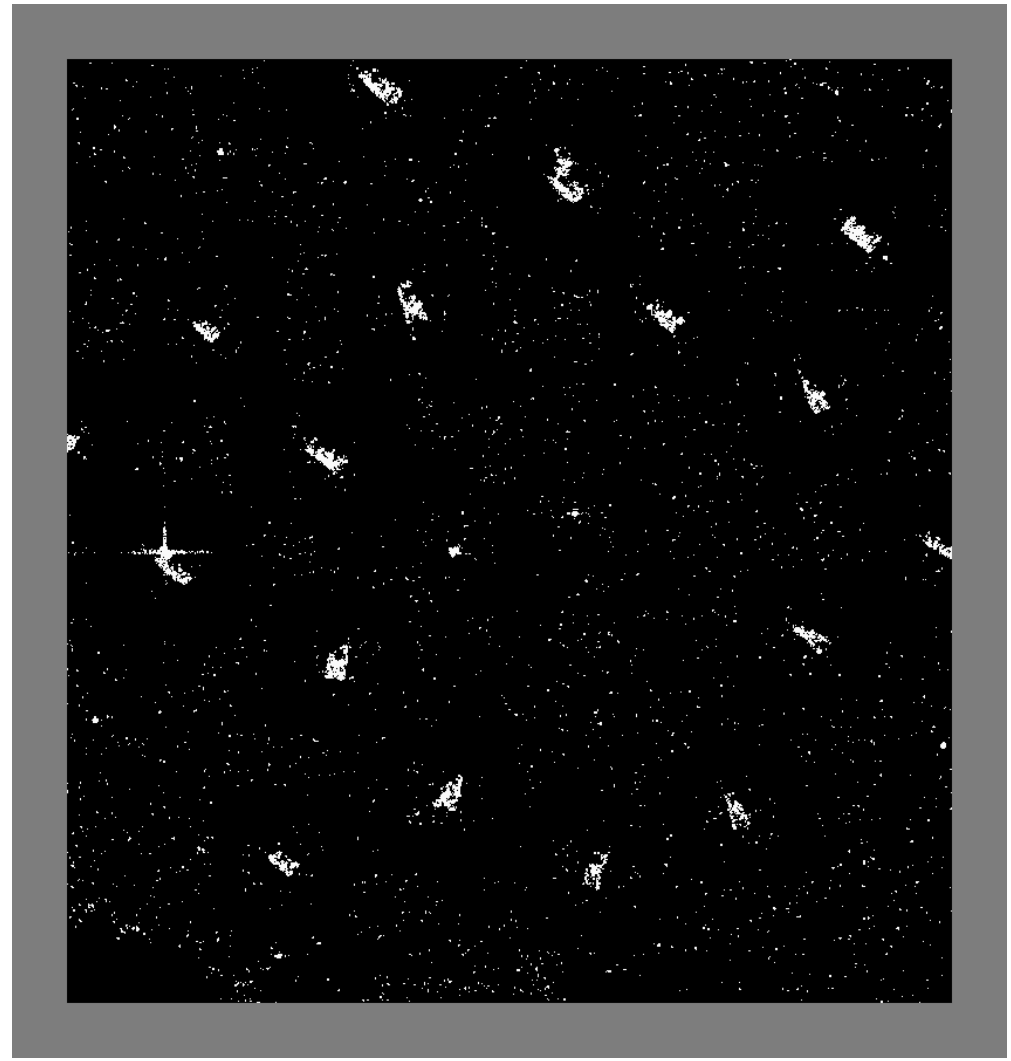
T-72 Tank

SAR Image Segmentation

- Application Domain: Surveillance using Synthetic Aperture Radar (SAR) Imagery
- Segmentation Task: identify regions of SAR images that are likely to contain "targets" (e.g. military vehicles, buildings, ...).
- Relation to Detection: Typical approach is to design a *pixel detector* that assigns a label 1 or 0 to every pixel in the image based on neighboring pixel values.
- Conventional Approach: look for pixels that are brighter than the background ... but the background intensity varies ... so use local data to estimate background statistics and implement a CFAR detector!

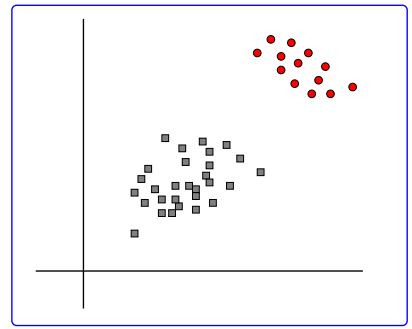
$\tau = 3$, False Alarm Rate under Gaussian = .0026, Actual Alarm Rate = .016

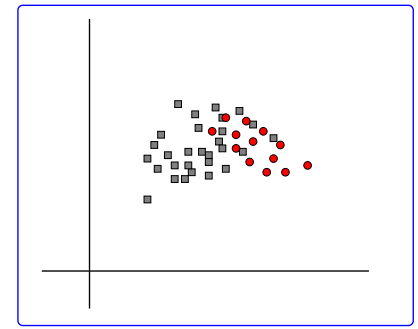




1-Class Summary

- common method in practice
- works well when there is a large separation between target and background





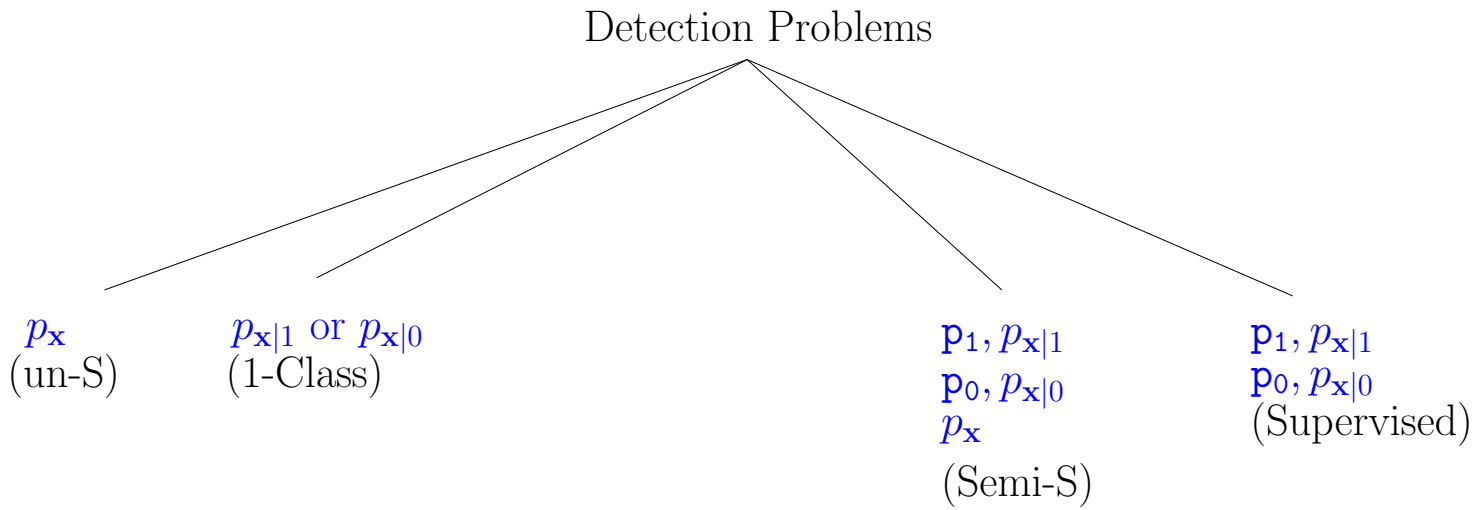
- feature selection (model selection, tuning, etc.) unclear
- most dangerous assumption: future targets drawn from the same distribution
- biggest weakness: ignores background distribution so
  - cannot *control* the "other" class error rate
  - cannot *validate* the overall error rate

A New Approach to Lopsided Detection

Basic Idea: incorporate information about the background distribution by using *deployment data*, i.e. by using *unlabeled* data gathered in the deployed environment.

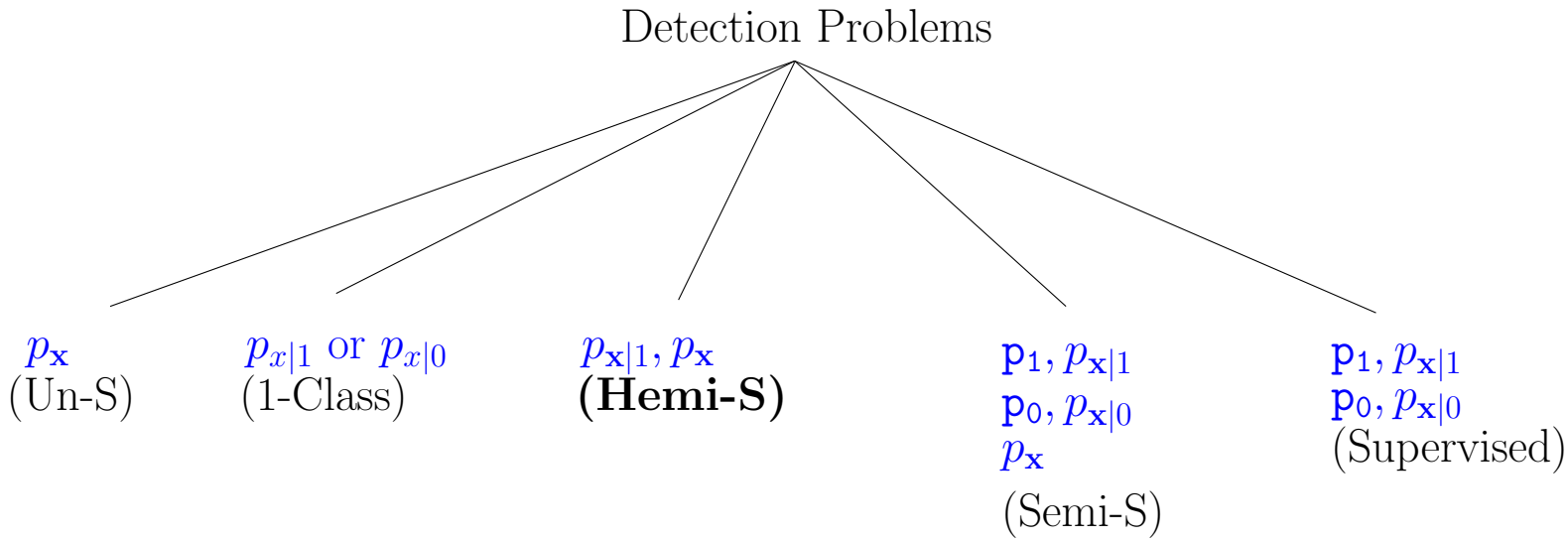
A New Approach to Lopsided Detection

Basic Idea: incorporate information about the background distribution by using *deployment data*, i.e. by using *unlabeled* data gathered in the deployed environment.



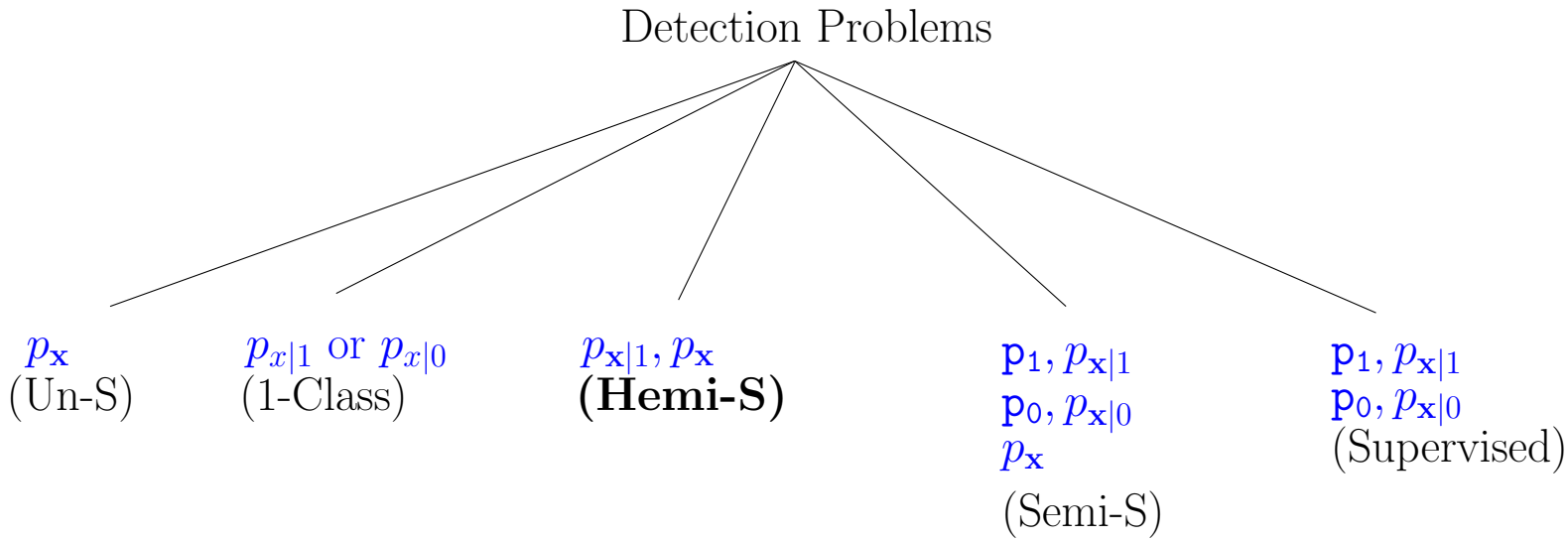
A New Approach to Lopsided Detection

Basic Idea: incorporate information about the background distribution by using *deployment data*, i.e. by using *unlabeled* data gathered in the deployed environment.



A New Approach to Lopsided Detection

Basic Idea: incorporate information about the background distribution by using *deployment data*, i.e. by using *unlabeled* data gathered in the deployed environment.



Hemi-Supervised Learning: Given target samples $(\mathbf{x}_1, ..., \mathbf{x}_{n_1}) \sim p_{\mathbf{x}|1}$, and (unlabeled) deployment samples $(\acute{\mathbf{x}}_1, ..., \acute{\mathbf{x}}_n) \sim p_{\mathbf{x}}$, design a detector $\hat{f}$ such that $e(\hat{f})$ is small.

Issues

- since deployment data contains a *mixture* of targets and background,
  - is there is enough information to control the error?
  - how do we use this data to design a detector?
  - how do we compare and validate potential solution methods?

- since unlabeled *design data = deployment data* we avoid the some of the concern over whether the design distribution is the same as the deployed distribution

- since the detector is designed *in the deployed environment* the method must be *fully automated, robust to distribution, and have guaranteed computational efficiency*

Error Decomposition

- Define the *alarm rate* of a detector f to be

$$a(f) := \int_{f(\mathbf{x})=1} p_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} \;=\; \mathtt{p}_1(1 - e_1(f)) \;+\; \mathtt{p}_0 e_0(f)$$

- Then the error rate is

$$\begin{aligned} e(f) &= \mathtt{p}_1 e_1(f) \;+\; \mathtt{p}_0 e_0(f) \\ &= \mathtt{p}_1 e_1(f) \;+\; \mathtt{p}_0 e_0(f) \;+\; a(f) \;-\; a(f) \\ &= \mathtt{p}_1 e_1(f) \;+\; \mathtt{p}_0 e_0(f) \;+\; a(f) \;-\; \mathtt{p}_1 \;+\; \mathtt{p}_1 e_1(f) \;-\; \mathtt{p}_0 e_0(f) \\ &= 2\mathtt{p}_1 e_1(f) \;+\; a(f) \;-\; \mathtt{p}_1 \end{aligned}$$

Error Decomposition

- Define the *alarm rate* of a detector f to be

$$a(f) := \int_{f(\mathbf{x})=1} p_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} \;=\; \mathtt{p}_1(1 - e_1(f)) \;+\; \mathtt{p}_0 e_0(f)$$

- Then the error rate is

$$\begin{aligned} e(f) &= \mathtt{p}_1 e_1(f) \;+\; \mathtt{p}_0 e_0(f) \\ &= \mathtt{p}_1 e_1(f) \;+\; \mathtt{p}_0 e_0(f) \;+\; a(f) \;-\; a(f) \\ &= \mathtt{p}_1 e_1(f) \;+\; \mathtt{p}_0 e_0(f) \;+\; a(f) \;-\; \mathtt{p}_1 \;+\; \mathtt{p}_1 e_1(f) \;-\; \mathtt{p}_0 e_0(f) \\ &= 2\mathtt{p}_1 e_1(f) \;+\; a(f) \;-\; \mathtt{p}_1 \\ &\approx 2\mathtt{p}_1 \frac{1}{n_1} \sum_{i=1}^{n_1} I(f(\mathbf{x}_i) = 0) \;+\; \frac{1}{n} \sum_{i=1}^{n} I(f(\acute{\mathbf{x}}_i) = 1) \;-\; \mathtt{p}_1 \end{aligned}$$

Consequences of Error Decomposition

If $\mathtt{p}_1$ is known

- even without labeled data from class 0 we can estimate the error rate (validation)
- we can design a classifier by solving a *surrogate supervised classification problem* with (weighted) training samples $(\bar{\mathbf{x}}_i, \bar{y}_i, \bar{w}_i)$ given by

$$(\bar{\mathbf{x}}_i, \bar{y}_i, \bar{w}_i) := \begin{cases} \left(\mathbf{x}_i, 1, \frac{2\mathtt{p}_1}{n_1(1+2\mathtt{p}_1)}\right), & \mathbf{x}_i \; labeled \\ \left(\acute{\mathbf{x}}_i, 0, \frac{1}{n(1+2\mathtt{p}_1)}\right), & \acute{\mathbf{x}}_i \; unlabeled \end{cases}$$

and surrogate error

$$\bar{e}(f) := \bar{\mathtt{p}}_1 \bar{e}_1(f) \; + \; \bar{\mathtt{p}}_0 \bar{e}_0(f) = \left(\frac{2\mathtt{p}_1}{1+2\mathtt{p}_1}\right) e_1(f) \; + \; \left(\frac{1}{1+2\mathtt{p}_1}\right) a(f)$$

- can use any classifier design method that is *fully automated, robust to distribution, and has guaranteed computational efficiency*
- allows a principled approach to feature selection (model selection, tuning, etc.)

Hemi-SVM Method

- *Hemi-SVM* algorithm with low order polynomial run-time guarantee (for all inputs) e.g. if $\bar{\mathtt{p}}_1 n > \bar{\mathtt{p}}_0 n_1$ and $\bar{n} := n + n_1$ then $O\left(d\bar{n}^2 + \frac{\bar{\mathtt{p}}_1^2}{\lambda\epsilon}\frac{\bar{n}^3}{n_1^2} + \bar{n}^2 \log \lambda \bar{\mathtt{p}}_1^2 \frac{n_1^2}{\bar{n}}\right)$

- guaranteed performance under mild distributional assumptions: e.g. for $n > n_1$ the excess error satisfies $e(\hat{f}) - e^* \leq c n_1^{-r}$ for $r \in (0,1)$ where e^* is the Bayes error.

Hemi-Supervised Summary

- error decomposition enables: validation, robust design methods, feature selection

- most dangerous assumptions:
  - p_1 is known
  - future (i.e. deployed) targets drawn from same distribution

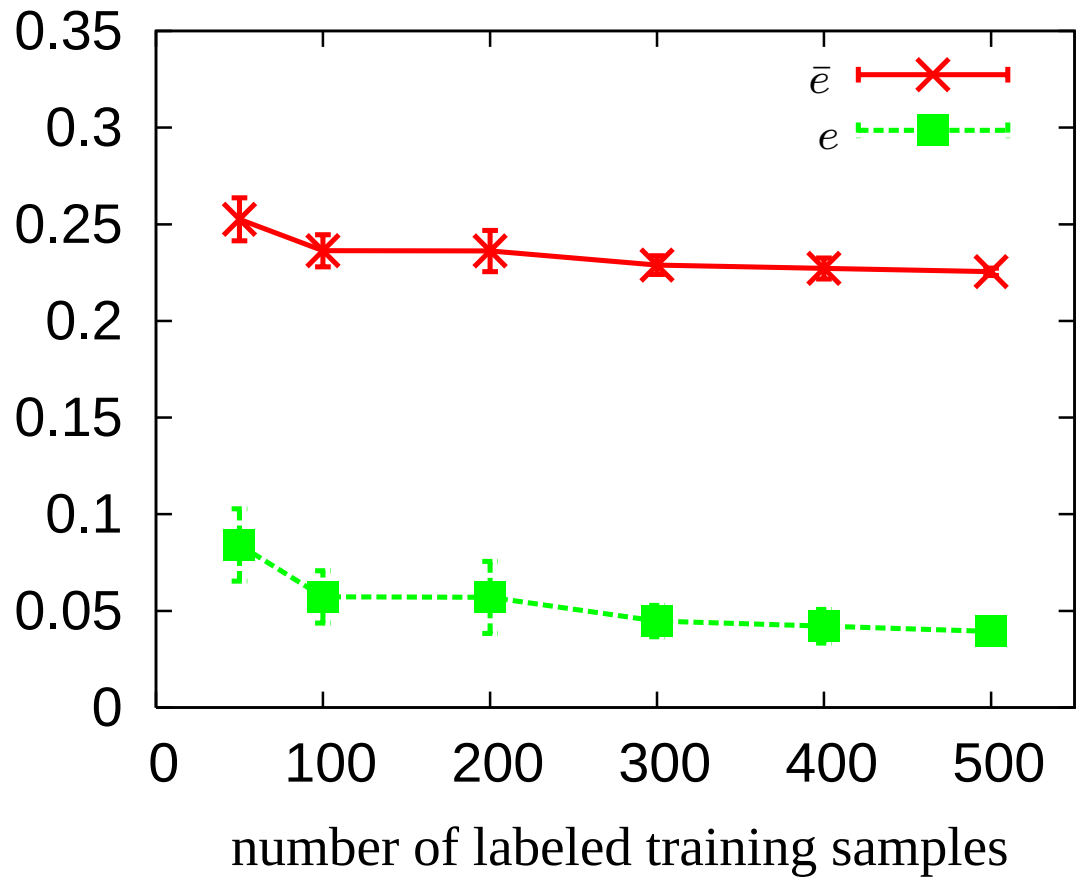
- How hard is it to estimate p_1?
  (differing opinions, we have a simple method ...)

- Can we develop a solution method that is robust to not-knowing p_1?

Experiments with simulated data
(where assumptions are satisfied)

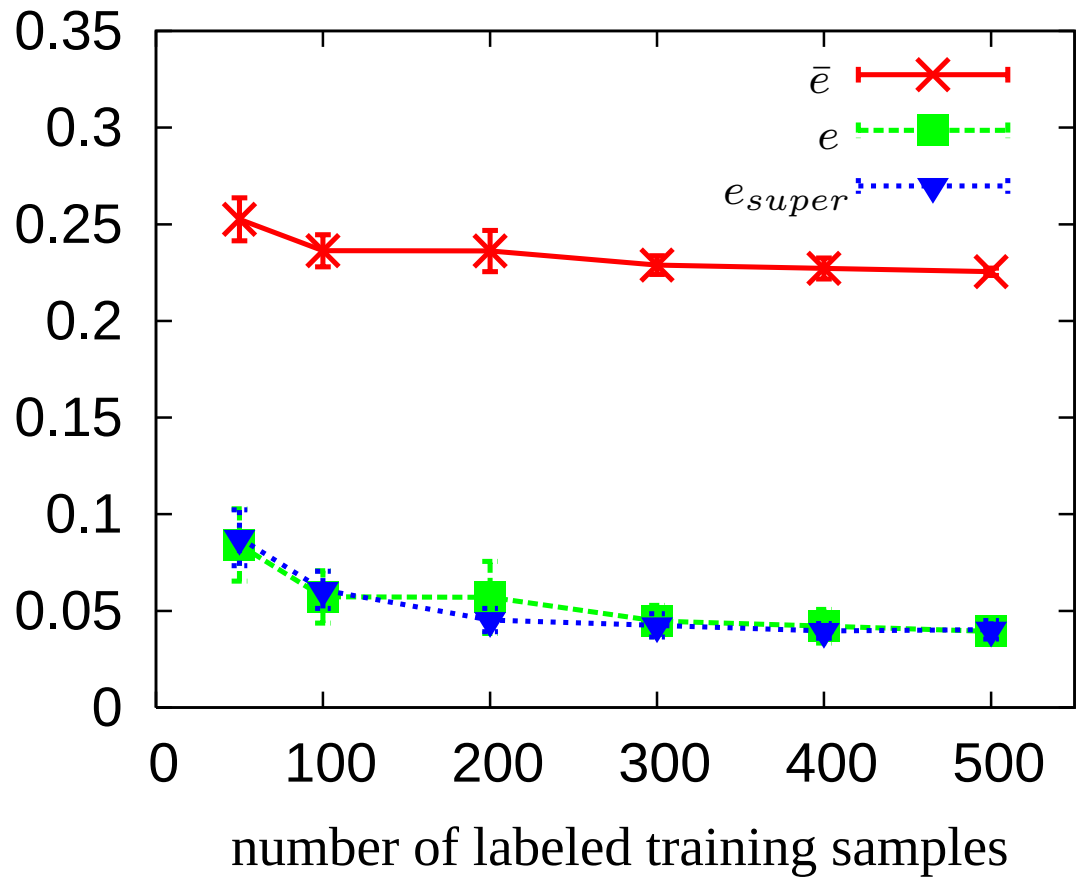
Hemi-Supervised Detection

(data dimension = 10, $\acute{n} = 1000$)



Hemi-Supervised Detection

(data dimension = 10, $\acute{n} = 1000$)



SAR Segmentation Revisited

The Hemi-Supervised SAR Segmenter

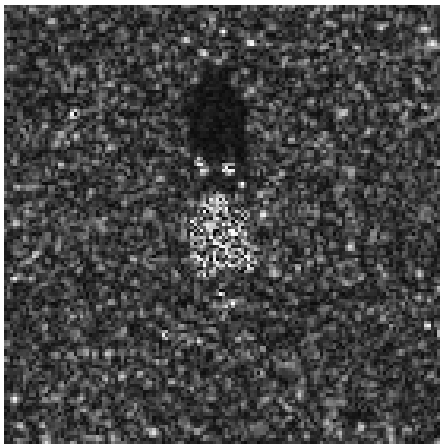
Relation to Lopsided Detection Problem: Abundance of target data available ahead of time, but the background data (clutter) is not available until deployment and is likely to be different

- Data Representation: Pixel values from a 10×10 window are used to predict the label for the center pixel. (T-72 tanks are roughly 45 pixels long and 25 pixels wide)
- Target Data: 274 (small) vehicle images of T-72 tanks (aspect angles uniformly distributed over the range 0 to 360 degrees)
- Deployment Data: Natural scenes with military vehicles
- For Comparison: Choose the threshold τ in the CFAR detector to minimize $\bar{e}$.

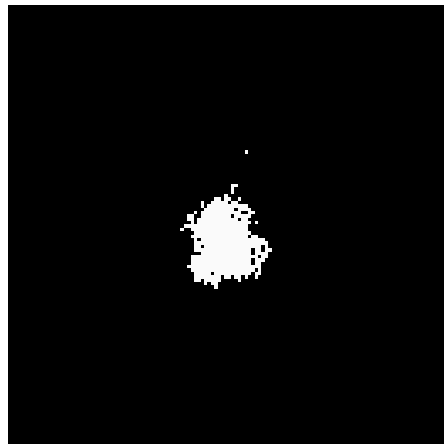
SAR Detector Results

p_1 estimate: .01 (prior lower and upper bounds: .003,.019)

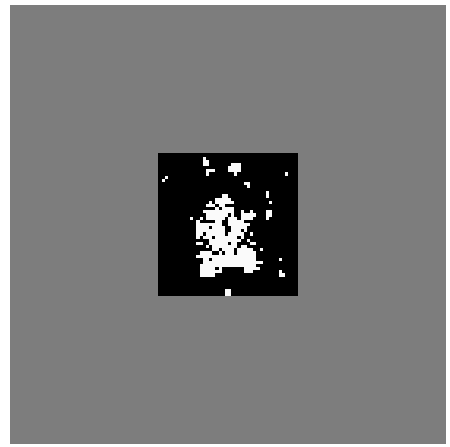
	Hemi	CFAR $\tau = 3$	CFAR $\tau^* = 6.46$
$\bar{e}$	.015	.025	.017
missed detection rate	.36	.44	.65
alarm rate	.0079	.016	.0038



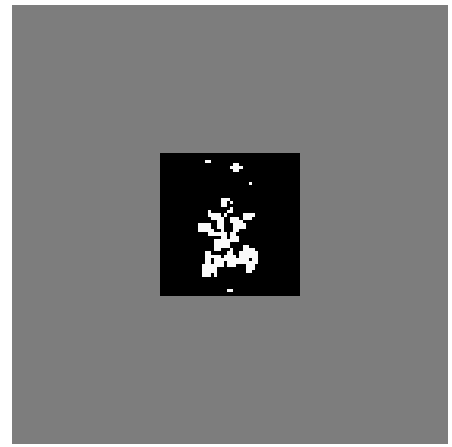
SAR Image



Hemi Label

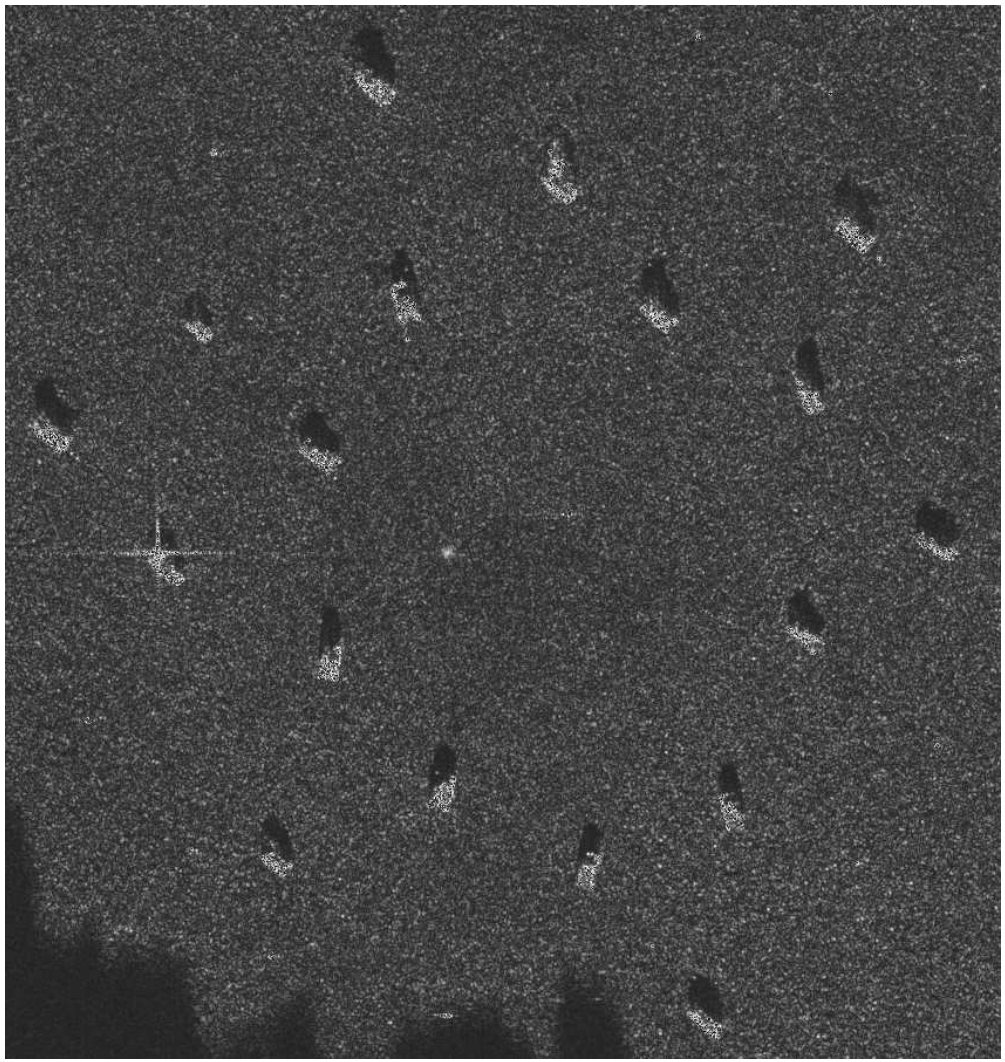


CFAR: $\tau = 3$

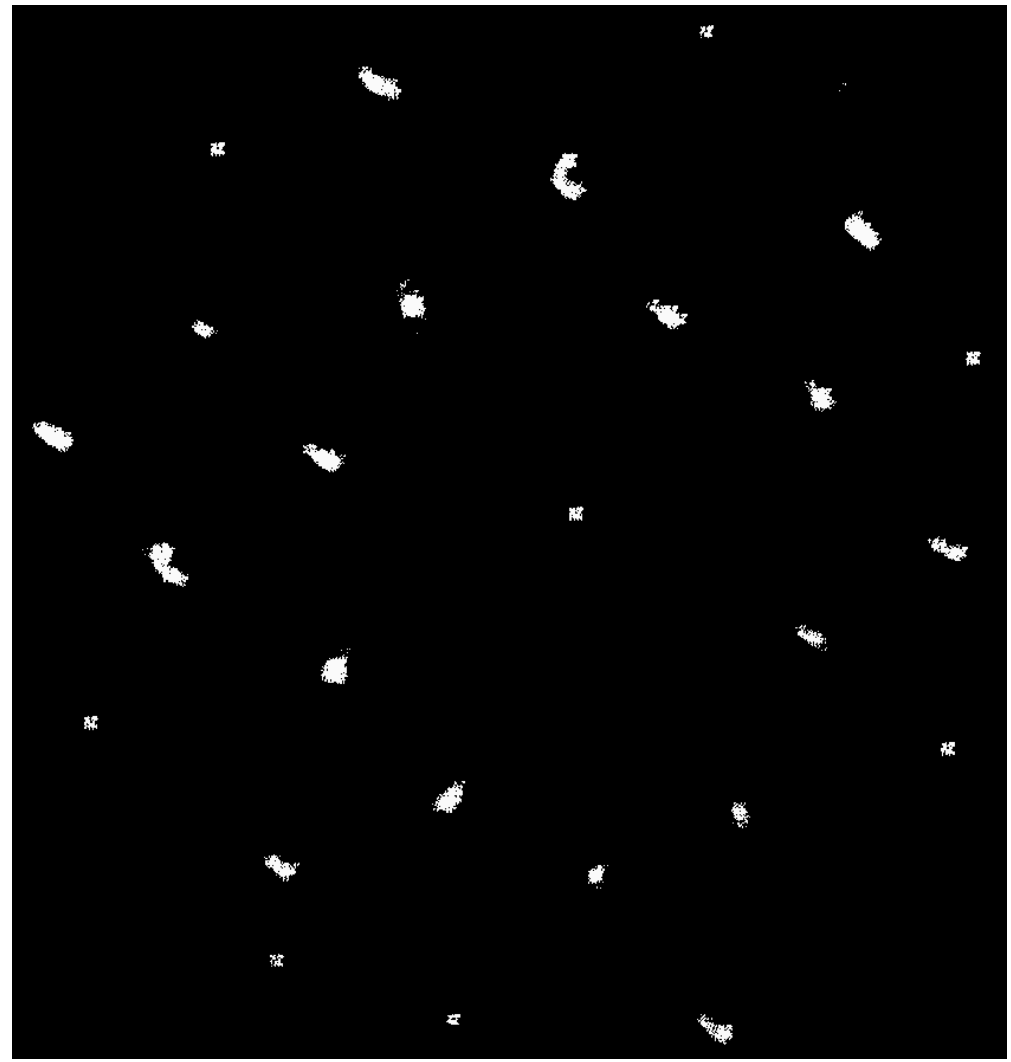


CFAR $\tau^* = 6.46$

SAR Detector Results

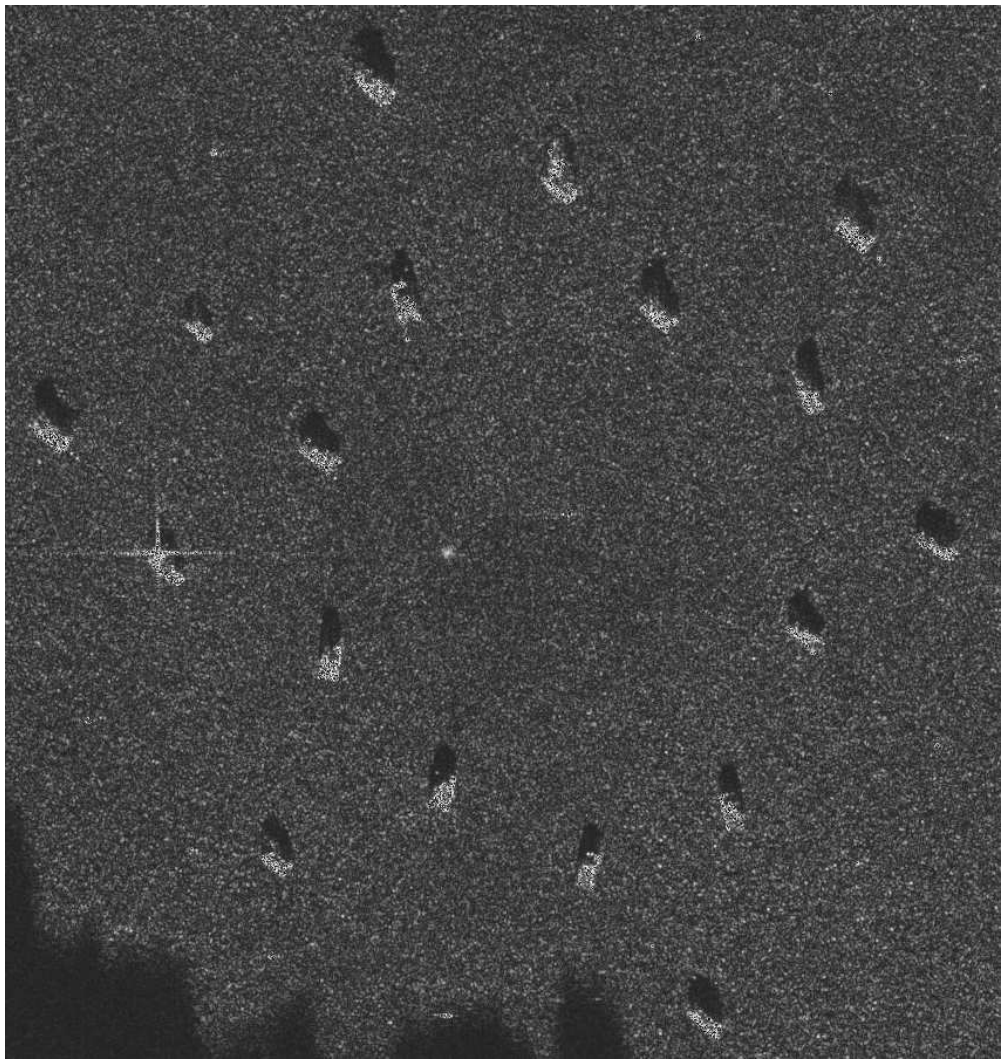


SAR Image

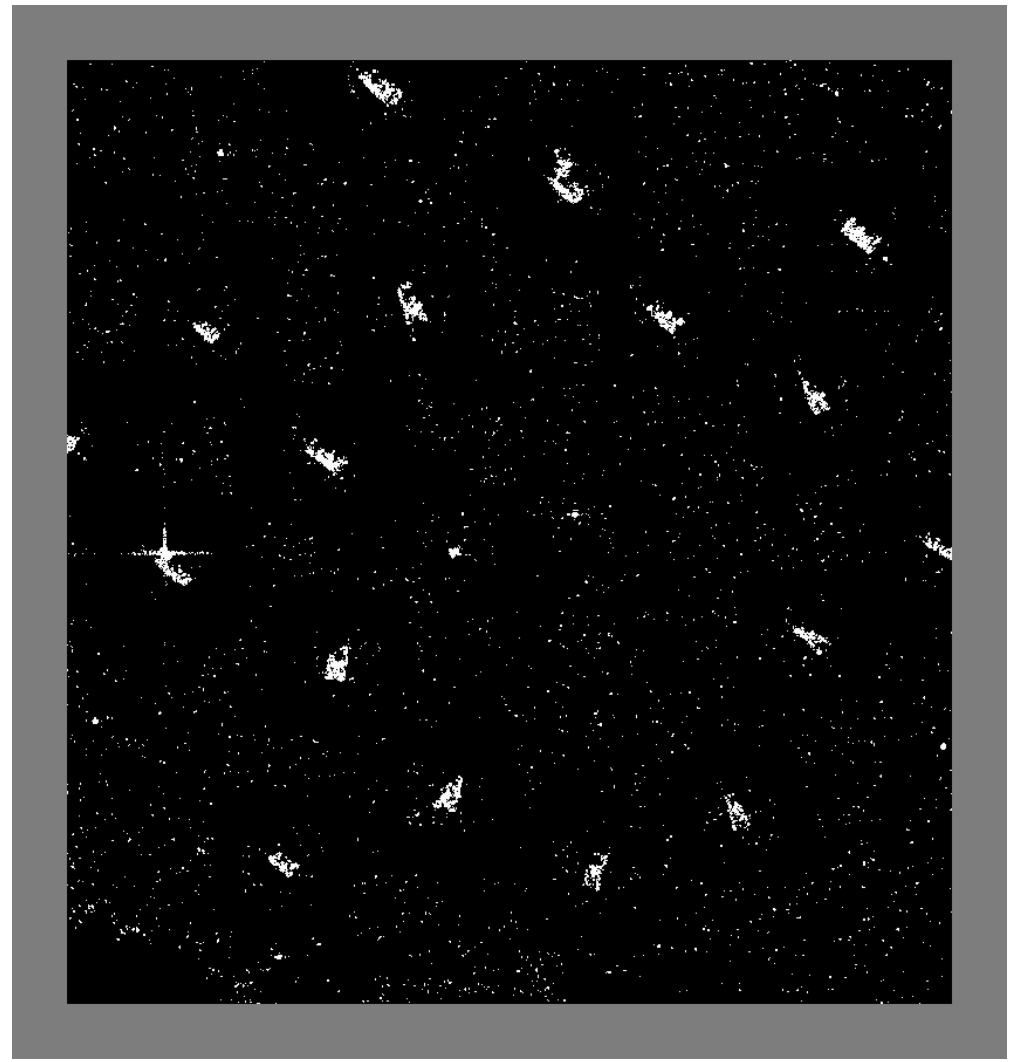


Hemi Label

SAR Detector Results

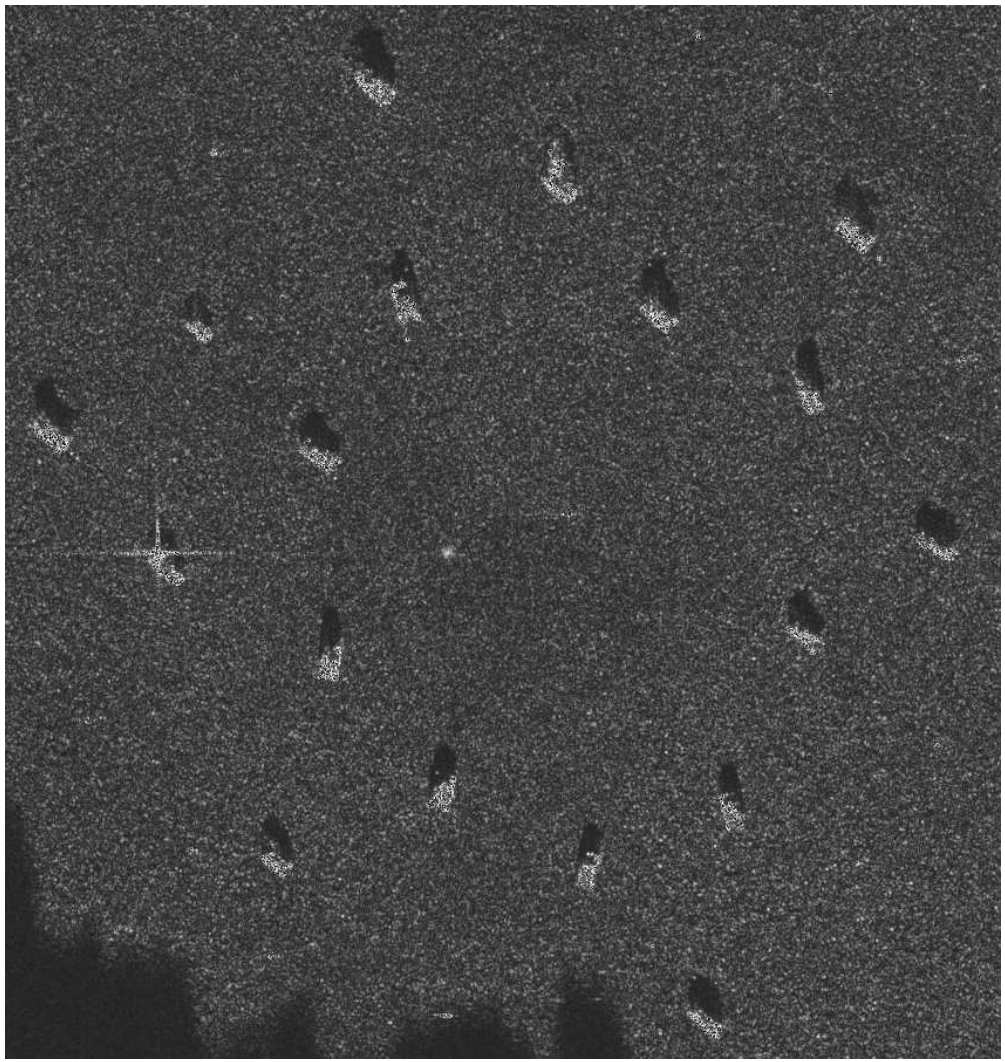


SAR Image

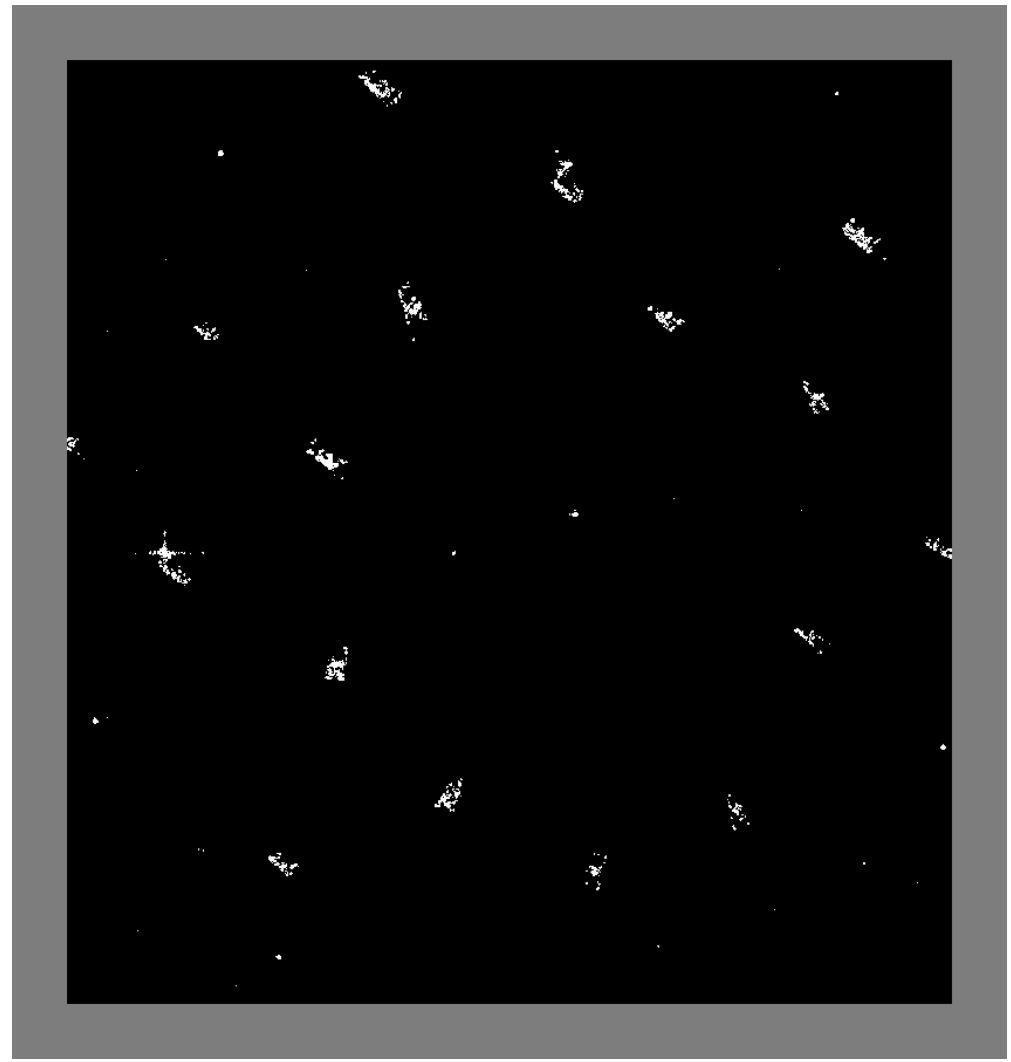


CFAR, $\tau = 3$

SAR Detector Results



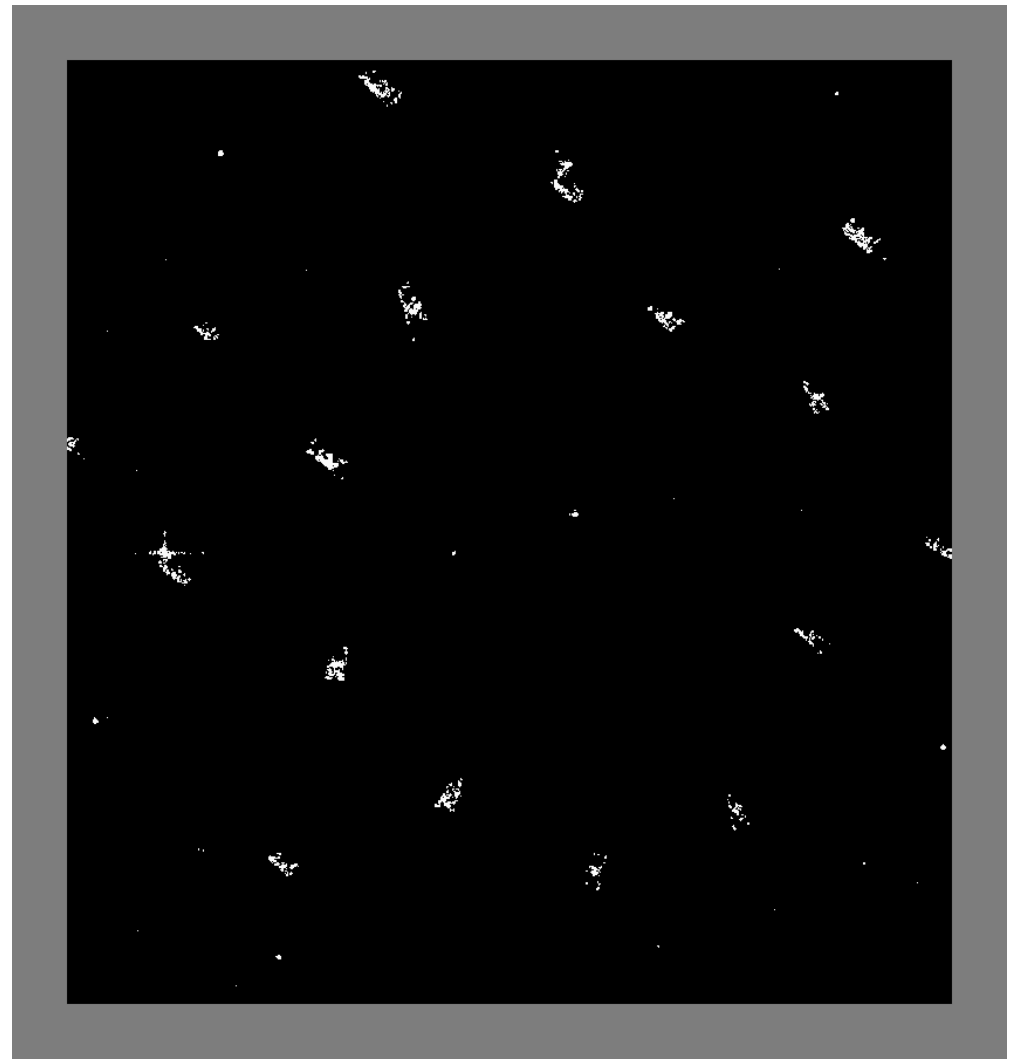
SAR Image



CFAR, $\tau^* = 6.46$

SAR Detector Results

Hemi Label



CFAR, $\tau^* = 6.46$

Other Hemi Solution Methods

- Iterative Labeling: Iterate the following two steps:
  1. use a discriminant function to identify a subset of the unlabeled samples that are considered most likely to be background
  2. apply a supervised classification method to the target and “predicted background” samples to obtain a new discriminant function
- Weighted Classification: various heuristics for the sample weights

Other Hemi Solution Methods

- Iterative Labeling: Iterate the following two steps:
  1. use a discriminant function to identify a subset of the unlabeled samples that are considered most likely to be background
  2. apply a supervised classification method to the target and "predicted background" samples to obtain a new discriminant function

- Weighted Classification: various heuristics for the sample weights

Observation: $f^*(\mathbf{x}) = \left[P_{1|\mathbf{x}}(\mathbf{x}) - 0.5\right]_0^1 = \left[\frac{\mathrm{p}_1 p_{\mathbf{x}|1}(\mathbf{x})}{p_{\mathbf{x}}(\mathbf{x})} - 0.5\right]_0^1$

Other Hemi Solution Methods

- Iterative Labeling: Iterate the following two steps:
  1. use a discriminant function to identify a subset of the unlabeled samples that are considered most likely to be background
  2. apply a supervised classification method to the target and "predicted background" samples to obtain a new discriminant function
- Weighted Classification: various heuristics for the sample weights

Observation: $f^*(\mathbf{x}) = \left[P_{1|\mathbf{x}}(\mathbf{x}) - 0.5\right]_0^1 = \left[\frac{\mathrm{p}_1 p_{\mathbf{x}|1}(\mathbf{x})}{p_{\mathbf{x}}(\mathbf{x})} - 0.5\right]_0^1$

- ML Estimates of $P_{1|\mathbf{x}}$: modifications of logistic regression (EM-type algorithms)
- Density Estimation: express f^* as one of several probability decompositions and then use a modification of the EM algorithm for mixtures of Gaussians to obtain ML estimates

Background/History/Lit. Review

- Naming:

  Hemi-supervised Learning

  Learning from [Only] Positive and Unlabeled Data (LPU)

  Positive Example Learning (POSEX)

  Positive Example Based Learning (PEBL)

- Selected References:
  - Steinberg & Cardell (1992): ML estimate of $P_{1|\mathbf{x}}$ (assume $\mathtt{p}_1$ known)
  - Dennis et. al. (1998, 2002, 2005): PAC framework, text experiments, importance of $\mathtt{p}_1$
  - Lee, Lin, Liu et al. (2003, 2005): ML estimate of $P_{1|\mathbf{x}}$, weighted classification, text experiments*
  - Zhang et. al. (2005,2008): ML estimate of $P_{1|\mathbf{x}}$, weighted classification, survey*
  - Yu et. al. (2004, 2006): iterative labeling*
  - Wang, et al. (2006): iterative labeling*
  - Elkan & Noto (2008): ML estimate of $P_{1|\mathbf{x}}$, weighted classification, ($\mathtt{p}_1 \sim$ different stat model)
  - Ward, Hastie et. al. (2009): ML estimate of $P_{1|\mathbf{x}}$, *claims* $\mathtt{p}_1$ *cannot be estimated*

* not clear how $\mathtt{p}_1$ is handled

Comparisons

Previous Methods:

- heuristics
- plug-in rules
- cascading estimates (high variance)
- coupled sample plan
- no validation method
- not ready for deployment (automation & robustness)

Methods Based on Error Decomposition:

- simple & direct
- the only cascaded estimate is p_1
- de-coupled sample plan
- validation (in deployed environment)
- feature selection, etc.
- almost ready for deployment ... need reliable estimate of p_1

Detector Design Paradigms

1-Class	Hemi	Semi	Supervised
$p_{\mathbf{x}\|1}$	$p_{\mathbf{x}}$ $p_{\mathbf{x}\|1}$	$p_{\mathbf{x}}$ $p_{\mathbf{x},y}$	$p_{\mathbf{x},y}$

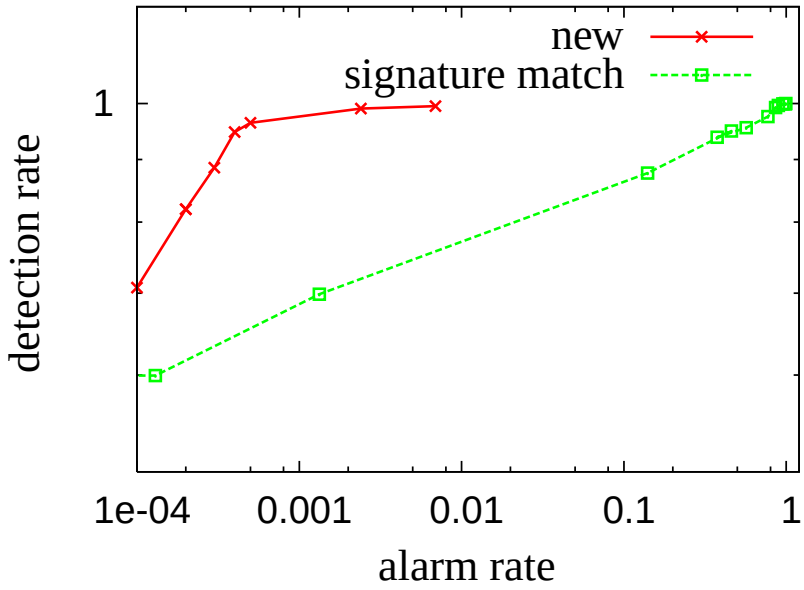
	1-Class	Hemi	Semi	Supervised
full error rate control		x	x	x
learn from unlabeled deployment data		x	x	

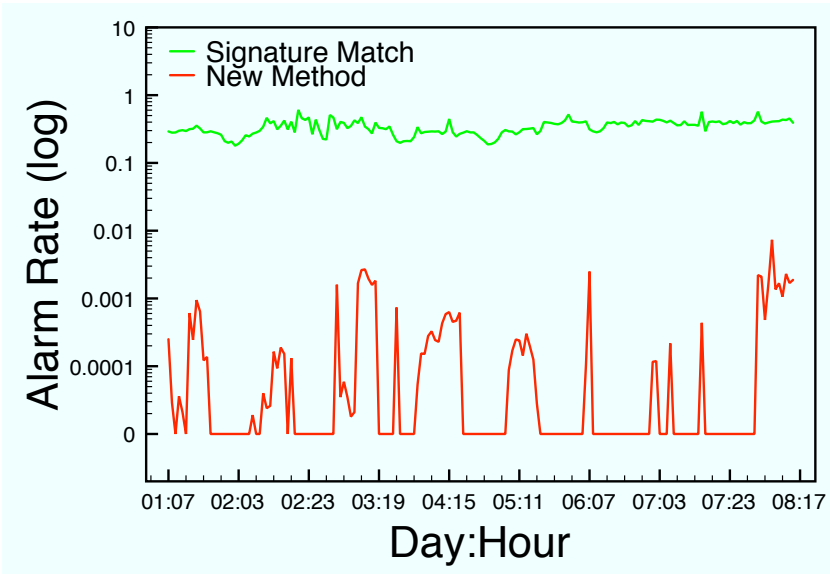
Detector Design Paradigms

1-Class	Hemi	Semi	Supervised
$p_{\mathbf{x}\|1}$	$p_{\mathbf{x}}$ $p_{\mathbf{x}\|1}$	$p_{\mathbf{x}}$ $p_{\mathbf{x},y}$	$p_{\mathbf{x},y}$

	1-Class	Hemi	Semi	Supervised
full error rate control		x	x	x
learn from unlabeled deployment data		x	x	
robust to differences between pre and post deployment distributions	2	1	3	4

Network Monitoring for Cybersecurity





- Problem: detect *CHAT* in *encrypted* network traffic
- Challenges:
  - limited information (due to encryption)
  - validating the deployed error rate
  - changing statistics (traffic patterns)
- Resolutions:
  - network traffic *meta-data*:

Packet Sizes	132, -122, 43, 28, -27, 23
Wait Times	-0.081, 0.003, -0.183, 0.002

  - **target** = CHAT meta-data from unencrypted traffic
  - **unlabeled** = ALL meta-data from encrypted traffic
  - **adaptive solution:** design a new (hemi) detector every hour

Summary

- lopsided detection problems are common
- 1-Class vs Hemi-LPU-POSEX-PEBL (vs Semi vs Supervised)
- Hemi error decomposition
  - validation in deployed environment
  - direct solution methods
- robustness to distributional assumptions:
  - use of unlabeled deployment data in design process
  - design in deployed environment $\Rightarrow$ greater demand on design method
- Example Applications: SAR, Cyber

Open/Other:

- estimating p_1 ... or *robust* Hemi
- NP Hemi (new paper out soon)
- algorithms and analysis for the Gaussian case (some surprises here)
- relation to other problems (content-based search, CFAR, Semi, ...)

Constant False Alarm Rate (CFAR) Detector

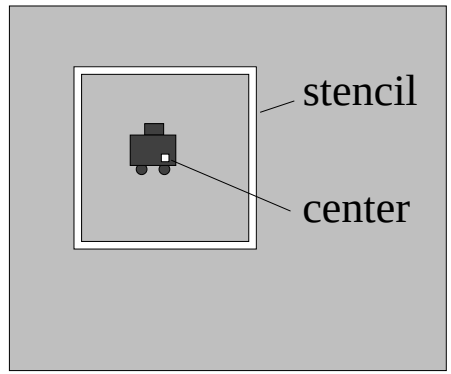
Assumptions:

- $\acute{p}_{\mathbf{x}|0}$ is *fixed locally* (but may vary in different regions of the image)
- $\acute{p}_{\mathbf{x}|0}$ is *Gaussian* (with different parameters in different regions)
- Target pixel values are generally *brighter* than background pixel values
- Gaussian parameters are not known, but can be estimated locally

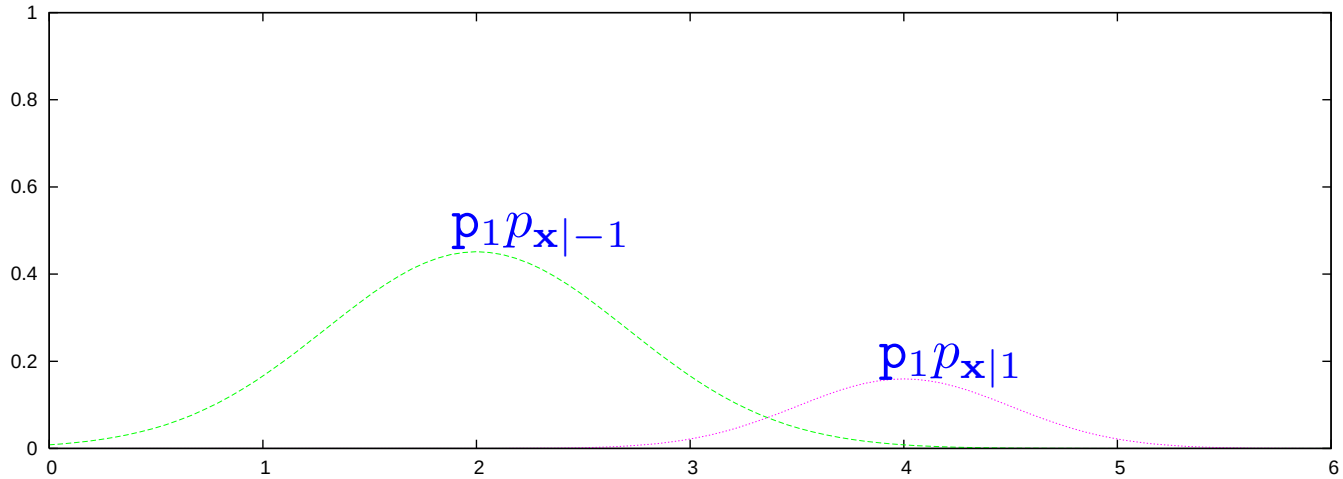
CFAR Detector: At pixel location (i, j) in the image

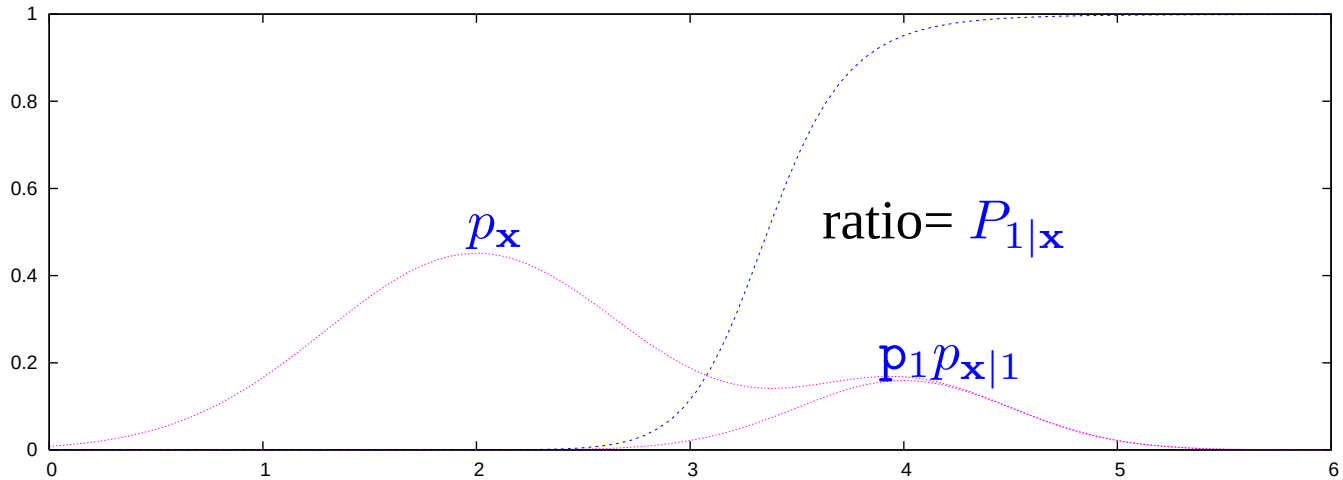
$$f(i,j) = Step\left[\frac{x(i,j) - \hat{\mu}(i,j)}{\hat{\sigma}(i,j)} - \tau\right]$$

- Cell Averaging (CA) CFAR: $\hat{\mu}(i,j)$ and $\hat{\sigma}(i,j)$ are computed using stencil region:



return


1
0.8
0.6
0.4
0.2
0
0
1
2
3
4
5
6
$\mathbf{p}_1 p_{\mathbf{x}|-1}$
$\mathbf{p}_1 p_{\mathbf{x}|1}$

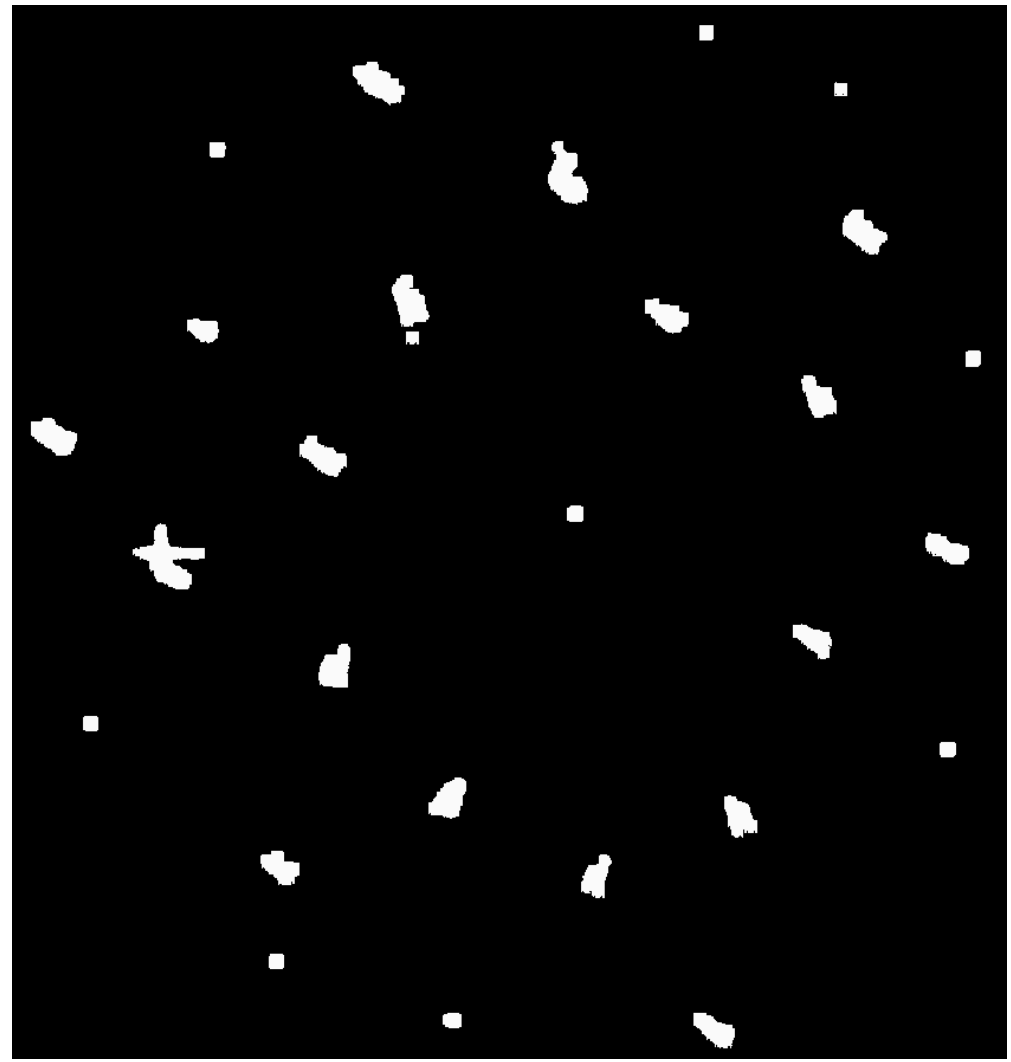

1
0.8
0.6
0.4
0.2
0
0
1
2
3
4
5
6
$p_{\mathbf{x}}$
ratio= $P_{1|\mathbf{x}}$
$\mathbf{p}_1 p_{\mathbf{x}|1}$

return

SAR Detector Results

SVM-Hemi Label

return



GML-Hemi Label