

# By def·i·ni·tion undefined

Adventures in anomaly (and anomalous change) detection

James Theiler

Intelligence and Space Research (ISR) Division  
Los Alamos National Laboratory



whispers

Research supported by the United States Department of Energy  
through the Los Alamos Laboratory Directed Research and Development (LDRD) Program.

# Outline: what I want to do in this talk

- Acknowledge colleagues (and promote our recent paper [1])
- Anomaly detection: a ridiculously vague concept
- RX: a tale of two derivations
- Kernels: an old dog learns a new trick
- Change: because everything is different

[1] S. Matteoli, M. Diani, JT. "An overview of background modeling for detection of targets and anomalies in hyperspectral remotely sensed imagery." *JSTARS* 7 (2014).

# Outline: what I want to do in this talk

- Anomaly detection: a ridiculously vague concept
- RX: a tale of two derivations
- Kernels: an old dog learns a new trick
- Change: because everything is different

# Outline: what I want to do in this talk

- Anomaly detection: a ridiculously vague concept
- RX: a tale of two derivations
- Kernels: an old dog learns a new trick
- Change: because everything is different



# Anomalies are defined by...

## Anomalies are defined by . . . what they are not

- “There is not an unambiguous way to define an anomaly, . . . an observation that deviates in some way from the background clutter.” – Matteoli *et al.* (2010)
- “Thus the multiplicity of possible spectra associated with the objects of interest and the complications of atmospheric compensation . . . detectors that seek to distinguish observations of unusual materials from typical background materials **without reference to target signatures or target subspaces**. . . Anomalies are defined with reference to a model of the background.” – Stein *et al.* (2002)
- “The basis of an anomaly detection system is accurate background characterization.” – Ashton (1999)

# Anomaly detection is defined by...

target detection when you don't know what the target is

# Anomaly detection is defined by... an advertisement from the 1970's

target detection when you don't know what the target is  
or

“What are you hungry for when you don't know what you're hungry for?”

# Anomaly detection is defined by... an advertisement from the 1970's

target detection when you don't know what the target is  
or

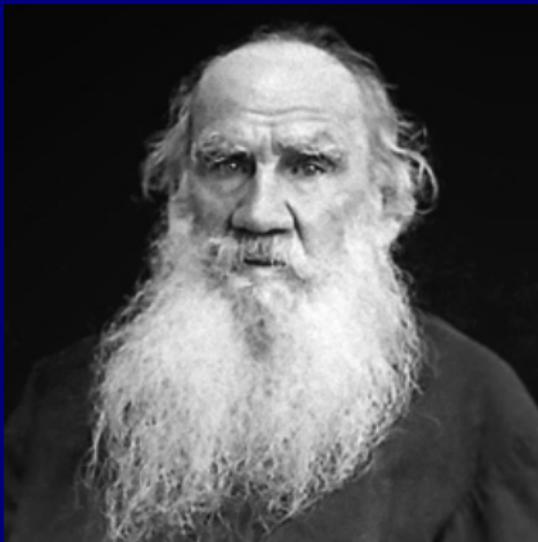
“What are you hungry for when you don't know what you're hungry for?”



“Something on a crisp Ritz cracker!”

## Anomalies are defined by... a novel from the 1870's

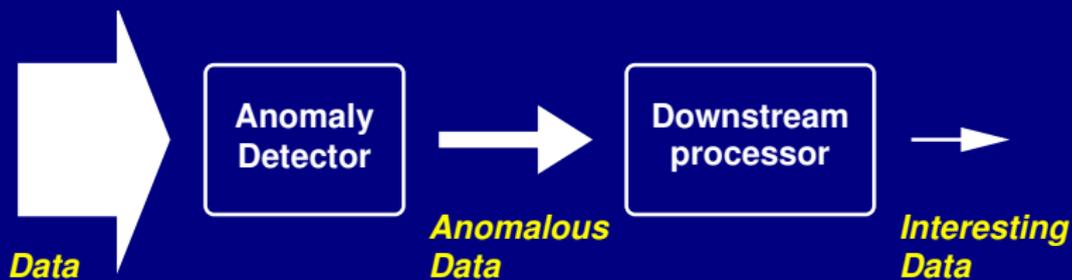
“All happy families are alike; every unhappy family is unhappy in its own way.”



Leo Tolstoy

# Anomalies are defined by... concepts that are even harder to define

- We really want *interesting* data, whatever that means
- Anomalies are “*potentially interesting*”
  - rare
  - unlike most of the data
- So anomaly *detection* is an intermediate triage







Definition

**RX**

Evaluate

Kernels

K-2d

K-1d

Change



# RX

# RX

I. S. Reed and X. Yu. "Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution." *IEEE Trans. Acoustics, Speech and Signal Processing* **38** (1990) 1760–1770.

# RX

I. S. Reed and X. Yu. "Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution." *IEEE Trans. Acoustics, Speech and Signal Processing* **38** (1990) 1760–1770.

P. C. Mahalanobis. "On the generalised distance in statistics," *Proc. National Institute of Sciences of India* **2** (1936) 49–55.

# RX: a twice-told tale

## ■ Derivation #1: using additive unknown target and GLRT

- $H_0: \mathbf{x} = \mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, C)$
- $H_1: \mathbf{x} = \mathbf{z} + \mathbf{t}$  for unknown  $\mathbf{t}$

## ■ Derivation #2: using explicit model for anomalies

- $H_0: \mathbf{x} = \mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, C)$
- $H_1: \mathbf{x} = \mathbf{t} \sim \mathcal{U}$

# RX: a twice-told tale

## ■ Derivation #1: using additive unknown target and GLRT

- $H_0: \mathbf{x} = \mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, C)$
- $H_1: \mathbf{x} = \mathbf{z} + \mathbf{t}$  for unknown  $\mathbf{t}$
- *Generalized* likelihood ratio:

$$\frac{\max_{\mathbf{t}} \exp \left[ -(\mathbf{x} - \mathbf{t} - \boldsymbol{\mu})^T C^{-1} (\mathbf{x} - \mathbf{t} - \boldsymbol{\mu}) / 2 \right]}{\exp \left[ -(\mathbf{x} - \boldsymbol{\mu})^T C^{-1} (\mathbf{x} - \boldsymbol{\mu}) / 2 \right]}$$

## ■ Derivation #2: using explicit model for anomalies

- $H_0: \mathbf{x} = \mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, C)$
- $H_1: \mathbf{x} = \mathbf{t} \sim \mathcal{U}$
- Straight likelihood ratio:

$$\frac{1}{\exp \left[ -(\mathbf{x} - \boldsymbol{\mu})^T C^{-1} (\mathbf{x} - \boldsymbol{\mu}) / 2 \right]}$$

# RX: a twice-told tale

## ■ Derivation #1: using additive unknown target and GLRT

- $H_0: \mathbf{x} = \mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, C)$
- $H_1: \mathbf{x} = \mathbf{z} + \mathbf{t}$  for unknown  $\mathbf{t}$
- *Generalized* likelihood ratio:

$$\frac{\max_{\mathbf{t}} \exp \left[ -(\mathbf{x} - \mathbf{t} - \boldsymbol{\mu})^T C^{-1} (\mathbf{x} - \mathbf{t} - \boldsymbol{\mu}) / 2 \right]}{\exp \left[ -(\mathbf{x} - \boldsymbol{\mu})^T C^{-1} (\mathbf{x} - \boldsymbol{\mu}) / 2 \right]}$$

## ■ Derivation #2: using explicit model for anomalies

- $H_0: \mathbf{x} = \mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, C)$
- $H_1: \mathbf{x} = \mathbf{t} \sim \mathcal{U}$
- Straight likelihood ratio:

$$\frac{1}{\exp \left[ -(\mathbf{x} - \boldsymbol{\mu})^T C^{-1} (\mathbf{x} - \boldsymbol{\mu}) / 2 \right]}$$

## ■ Both derivations lead to $\mathcal{A}(\mathbf{x}) = (\mathbf{x} - \boldsymbol{\mu})^T C^{-1} (\mathbf{x} - \boldsymbol{\mu})$

# RX: a twice-told tale

## ■ Derivation #1: using additive unknown target and GLRT

- $H_0: \mathbf{x} = \mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, C)$
- $H_1: \mathbf{x} = \mathbf{z} + \mathbf{t}$  for unknown  $\mathbf{t}$  ←  $\mathbf{t}$  is “undefined”
- *Generalized* likelihood ratio:

$$\frac{\max_{\mathbf{t}} \exp \left[ -(\mathbf{x} - \mathbf{t} - \boldsymbol{\mu})^T C^{-1} (\mathbf{x} - \mathbf{t} - \boldsymbol{\mu}) / 2 \right]}{\exp \left[ -(\mathbf{x} - \boldsymbol{\mu})^T C^{-1} (\mathbf{x} - \boldsymbol{\mu}) / 2 \right]}$$

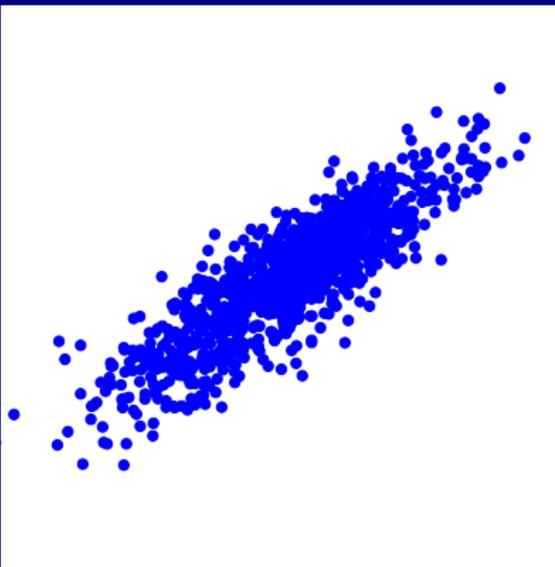
## ■ Derivation #2: using explicit model for anomalies

- $H_0: \mathbf{x} = \mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, C)$
- $H_1: \mathbf{x} = \mathbf{t} \sim \mathcal{U}$  ←  $\mathbf{t}$  is “defined”

- Straight likelihood ratio:  $\frac{1}{\exp \left[ -(\mathbf{x} - \boldsymbol{\mu})^T C^{-1} (\mathbf{x} - \boldsymbol{\mu}) / 2 \right]}$

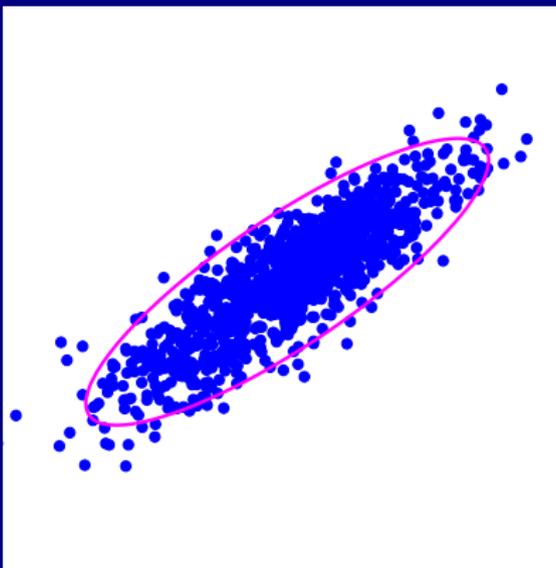
## ■ Both derivations lead to $\mathcal{A}(\mathbf{x}) = (\mathbf{x} - \boldsymbol{\mu})^T C^{-1} (\mathbf{x} - \boldsymbol{\mu})$

# RX as classification



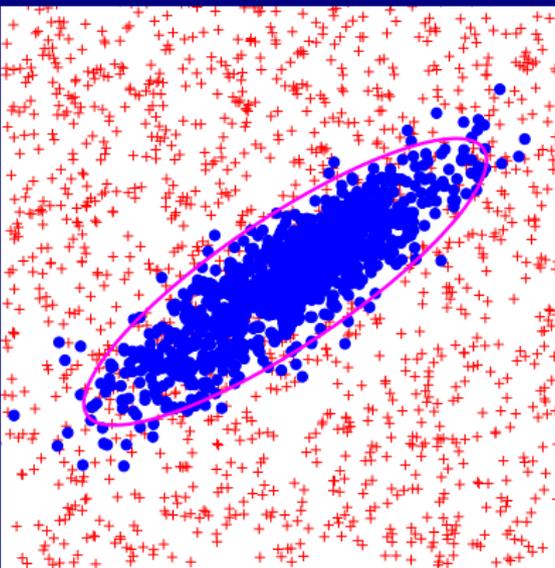
- Blue points: observed data

# RX as classification



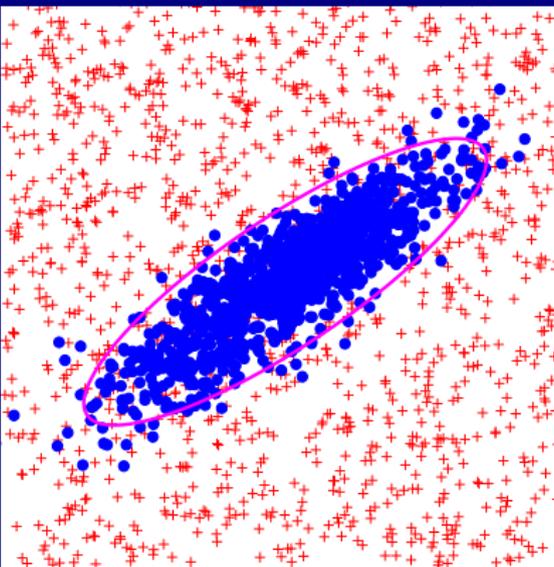
- Blue points: observed data
- Magenta line: contour of constant  $\mathcal{A}(\mathbf{x})$

# RX as classification



- Blue points: observed data
- Red crosses: artificial data
  - drawn from an *explicit* anomaly model:  $\mathbf{t} \sim \mathcal{U}$
- Magenta line: contour of constant  $\mathcal{A}(\mathbf{x})$

# RX as classification



- Blue points: observed data
- Red crosses: artificial data
  - drawn from an *explicit* anomaly model:  $\mathbf{t} \sim \mathcal{U}$
- Magenta line: contour of constant  $\mathcal{A}(\mathbf{x})$  separates the classes

# We've defined anomalies! $\mathbf{t} \sim \mathcal{U}$

Two consequences

1. Can use likelihood ratio to build optimal anomaly detectors for any background model (that provides a density function)

$$\mathcal{A}(\mathbf{x}) = \frac{\mathcal{P}(\mathbf{x} \text{ is anomalous})}{\mathcal{P}(\mathbf{x} \text{ is normal})} = \frac{U(\mathbf{x})}{p(\mathbf{x})} = \frac{1}{p(\mathbf{x})}$$

2. Can evaluate performance of anomaly detectors in an objective and unambiguous way

# Evaluate performance of anomaly detectors in an objective and unambiguous way

- Proper evaluation is given by detection and false alarm rates
  - But detection rate depends on choice of targets
  - (and those targets are by definition undefined?)

# Evaluate performance of anomaly detectors in an objective and unambiguous way

- Proper evaluation is given by detection and false alarm rates
  - But detection rate depends on choice of targets
  - (and those targets are by definition undefined?)
  
- Plan A: the ultimate arbiter of anomaly detection performance



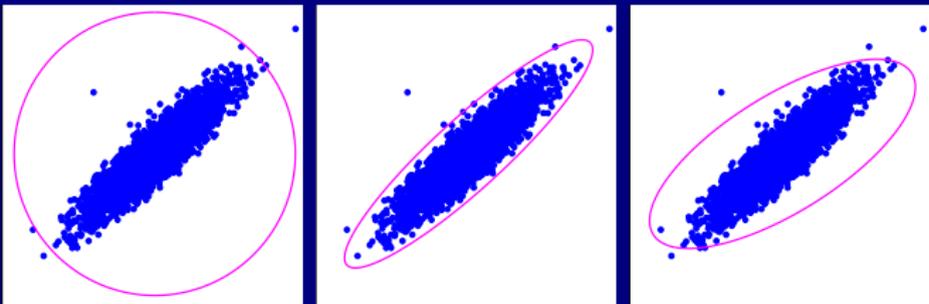
# Evaluate performance of anomaly detectors in an objective and unambiguous way

- Proper evaluation is given by detection and false alarm rates
  - But detection rate depends on choice of targets
  - (and those targets are by definition undefined?)
  
- Plan B: Implant targets using  $\mathbf{t} \sim \mathcal{U}$

# Evaluate performance of anomaly detectors in an objective and unambiguous way

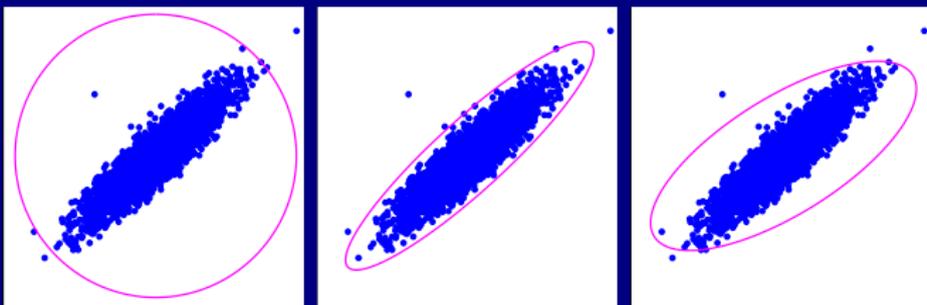
- Proper evaluation is given by detection and false alarm rates
  - But detection rate depends on choice of targets
  - (and those targets are by definition undefined?)
  
- Plan B: Implant targets using  $\mathbf{t} \sim \mathcal{U}$   
Equivalently: plot volume vs false alarm rate

# Volume vs false alarm rate



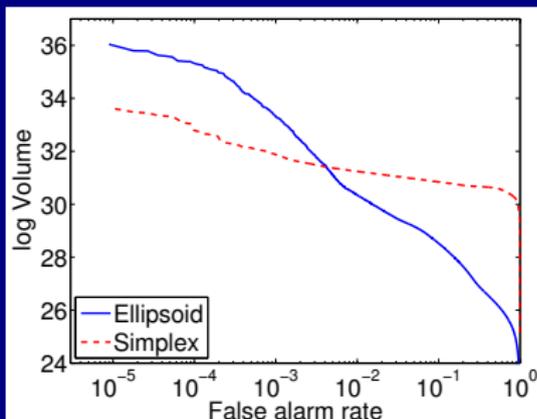
Three anomaly detectors with  $P_{fa} = 0.001$

# Volume vs false alarm rate

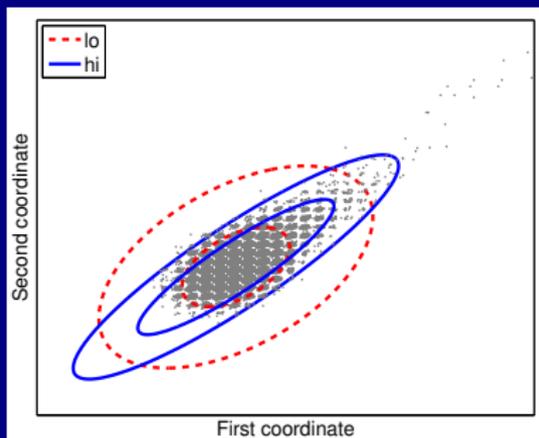


Three anomaly detectors with  $P_{fa} = 0.001$

- Volume grows with decreasing false alarm rate
- Volume is a proxy for missed detection rate
- Smaller volumes are better

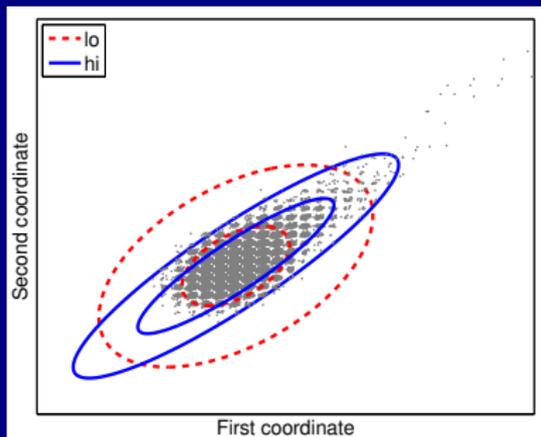


# Characterizing data on the periphery



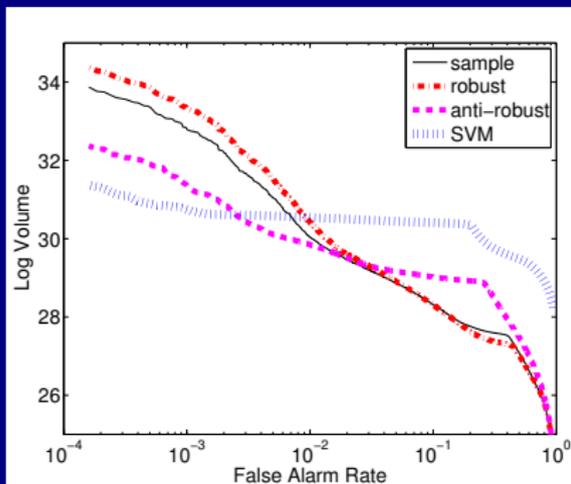
- AVIRIS data, first two channels
- Contours drawn for false alarm rates of 5% and 0.1%
- Red ellipses more effective at 5%
- Blue ellipses more effective at 0.1%

# Characterizing data on the periphery



- AVIRIS data, first two channels
- Contours drawn for false alarm rates of 5% and 0.1%
- Red ellipses more effective at 5%
- Blue ellipses more effective at 0.1%

- Evaluate performance for 224-channel image
- Plot log-volume vs false alarm rate
- Compare covariance matrices



# Some challenges with $\mathbf{t} \sim \mathcal{U}$

- computing volumes in high-dimensional spaces
- coordinate dependence
  - eg,  $\log \mathbf{t} \not\sim \mathcal{U}$
- dimension dependence: projections are problematic
  - eg, can't compare RX to SSRX
  - SSRX contours have infinite volume in original space
- non-probabilistic approaches:
  - eg, kernels, graphs, manifolds, sparse models, etc.
  - need contours of constant  $\mathcal{A}(\mathbf{x})$
  - need to be able to compute volume inside those contours





Definition

RX

Evaluate

**Kernels**

K-2d

K-1d

Change



# kernels

# kernels

with thanks to:

Guen Grosklos, Stefania Matteoli, Gustavo Camps-Valls, and Heesung Kwon

# kernels

with thanks to:

Guen Grosklos, Stefania Matteoli, Gustavo Camps-Valls, and Heesung Kwon  
work

# kernels

with thanks to:

Guen Grosklos, Stefania Matteoli, Gustavo Camps-Valls, and Heesung Kwon  
work talk

# kernels



non-Gaussian kernel

with thanks to:

Guen Groszklos, Stefania Matteoli, Gustavo Camps-Valls, and Heesung Kwon

work

talk

# Kernel Density Estimation (traditional)

- Given data:  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$
- Drawn from an *unknown* probability density function  $p(\mathbf{x})$
- Estimate  $p(\mathbf{x})$  from the data
  
- MLE:  $\hat{p}(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^N \delta(\mathbf{x} - \mathbf{x}_n)$
  
- Regularize  $\delta(\cdot)$  with kernel:  $\kappa(\mathbf{x}, \mathbf{x}_n) = c \cdot \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_n\|^2}{2\sigma^2}\right)$
  
- KDE:  $\hat{p}(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^N \kappa(\mathbf{x}, \mathbf{x}_n)$

# Kernel Density Estimation (traditional)

- Given data:  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$
- Drawn from an *unknown* probability density function  $p(\mathbf{x})$
- Estimate  $p(\mathbf{x})$  from the data
  
- MLE:  $\hat{p}(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^N \delta(\mathbf{x} - \mathbf{x}_n)$
  
- Regularize  $\delta()$  with kernel:  $\kappa(\mathbf{x}, \mathbf{x}_n) = c \cdot \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_n\|^2}{2\sigma^2}\right)$
  
- KDE:  $\hat{p}(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^N \kappa(\mathbf{x}, \mathbf{x}_n)$
  
- KDE anomaly detector:  $\mathcal{A}(\mathbf{x}) = 1 - \frac{1}{N} \sum_{n=1}^N \kappa(\mathbf{x}, \mathbf{x}_n)$

# Kernels (modern)

- Function  $\phi : \mathbb{R}^d \rightarrow F$  maps data to feature space
  - $\mathbf{x} \in \mathbb{R}^d$  is data in “real” space
  - $\phi(\mathbf{x}) \in F$  is data mapped to feature space
- Scalar dot products in feature space  $F$  can be expressed as functions of the values in real space.

$$\kappa(\mathbf{r}, \mathbf{s}) = \phi(\mathbf{r})^T \phi(\mathbf{s})$$

- *Trick*: Even though  $\phi$  is presumed to “exist” in some abstract philosophical/mathematical sense, we may not actually need to use it, as long as we have the kernel function  $\kappa$ .
- Gaussian kernel:

$$\kappa(\mathbf{r}, \mathbf{s}) = c \cdot \exp\left(-\frac{\|\mathbf{r} - \mathbf{s}\|^2}{2\sigma^2}\right)$$

# KDE anomaly detector, revisited

- Given data:  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$
- Map to feature space:  $\{\phi(\mathbf{x}_1), \dots, \phi(\mathbf{x}_N)\}$
- Define centroid:  $\boldsymbol{\mu}_\phi = \frac{1}{N} \sum_{n=1}^N \phi(\mathbf{x}_n)$
- Define anomalousness as distance to centroid in feature space:

$$\begin{aligned} \mathcal{A}(\mathbf{x}) &= \|\phi(\mathbf{x}) - \boldsymbol{\mu}_\phi\|^2 = (\phi(\mathbf{x}) - \boldsymbol{\mu}_\phi)^T (\phi(\mathbf{x}) - \boldsymbol{\mu}_\phi) \\ &= \underbrace{\phi(\mathbf{x})^T \phi(\mathbf{x})}_{\kappa(\mathbf{x}, \mathbf{x})=\text{constant}} - 2\phi(\mathbf{x})^T \boldsymbol{\mu}_\phi + \underbrace{\boldsymbol{\mu}_\phi^T \boldsymbol{\mu}_\phi}_{\text{constant}} \end{aligned}$$

- Note:  $\phi(\mathbf{x})^T \boldsymbol{\mu}_\phi = \frac{1}{N} \sum_{n=1}^N \phi(\mathbf{x})^T \phi(\mathbf{x}_n) = \frac{1}{N} \sum_{n=1}^N \kappa(\mathbf{x}, \mathbf{x}_n)$
- Leads to KDE anomaly detector:

$$\mathcal{A}(\mathbf{x}) = \text{constant} - \frac{2}{N} \sum_{n=1}^N \kappa(\mathbf{x}, \mathbf{x}_n)$$

# Support Vector Domain Description (SVDD)

- Given data:  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$
- Map to feature space:  $\{\phi(\mathbf{x}_1), \dots, \phi(\mathbf{x}_N)\}$
- Define *adaptive* centroid:  $\mathbf{a}_\phi = \sum_n a_n \phi(\mathbf{x}_n)$
- Minimize radius of sphere that encloses the data

$$\min_{R, \mathbf{a}_\phi} R^2$$

$$\text{subject to: } \|\phi(\mathbf{x}_n) - \mathbf{a}_\phi\|^2 \leq R^2$$

- Leads to SVDD anomaly detector

$$\mathcal{A}(\mathbf{x}) = \|\phi(\mathbf{x}) - \mathbf{a}_\phi\|^2 = \text{constant} - \sum_n a_n \kappa(\mathbf{x}, \mathbf{x}_n)$$

- Similar to KDE, but with unequal weights on the points  $\mathbf{x}_n$
- Points with  $a_n > 0$  are “support vectors”

# Support Vector Domain Description (SVDD)

- Given data:  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$
- Map to feature space:  $\{\phi(\mathbf{x}_1), \dots, \phi(\mathbf{x}_N)\}$
- Define *adaptive* centroid:  $\mathbf{a}_\phi = \sum_n a_n \phi(\mathbf{x}_n)$
- Minimize radius of sphere that encloses most of the data

$$\min_{R, \mathbf{a}_\phi, \xi} R^2 + c \sum_n \xi_n$$

$$\text{subject to: } \|\phi(\mathbf{x}_n) - \mathbf{a}_\phi\|^2 \leq R^2 + \xi_n$$

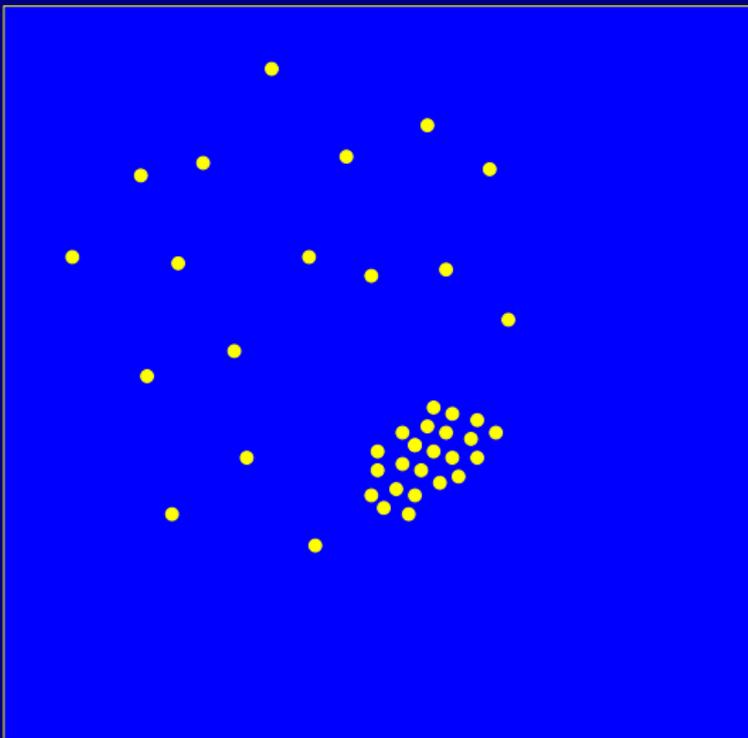
$$\text{and: } \xi_n \geq 0$$

- Leads to SVDD anomaly detector

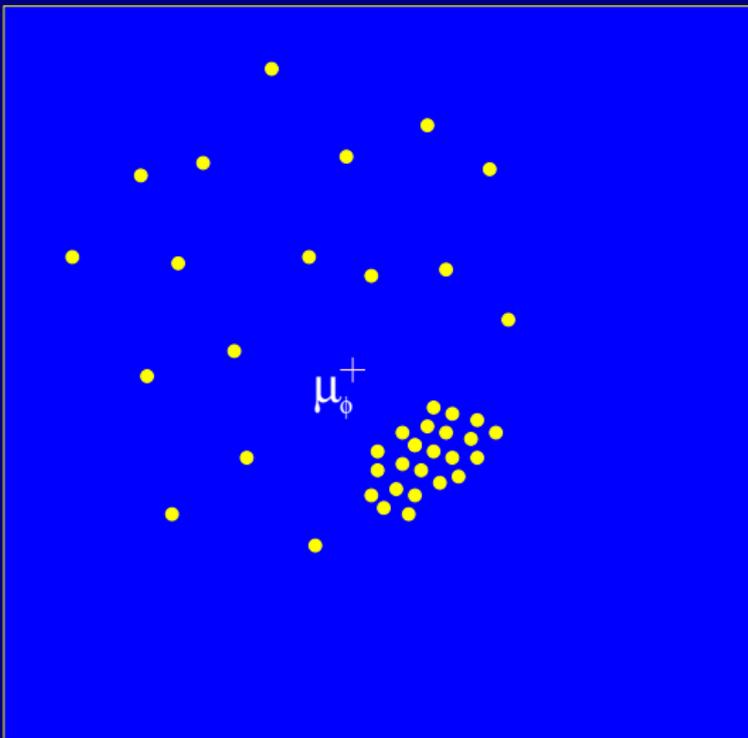
$$\mathcal{A}(\mathbf{x}) = \|\phi(\mathbf{x}) - \mathbf{a}_\phi\|^2 = \text{constant} - \sum_n a_n \kappa(\mathbf{x}, \mathbf{x}_n)$$

- Similar to KDE, but with unequal weights on the points  $\mathbf{x}_n$
- Points with  $a_n > 0$  are “support vectors”

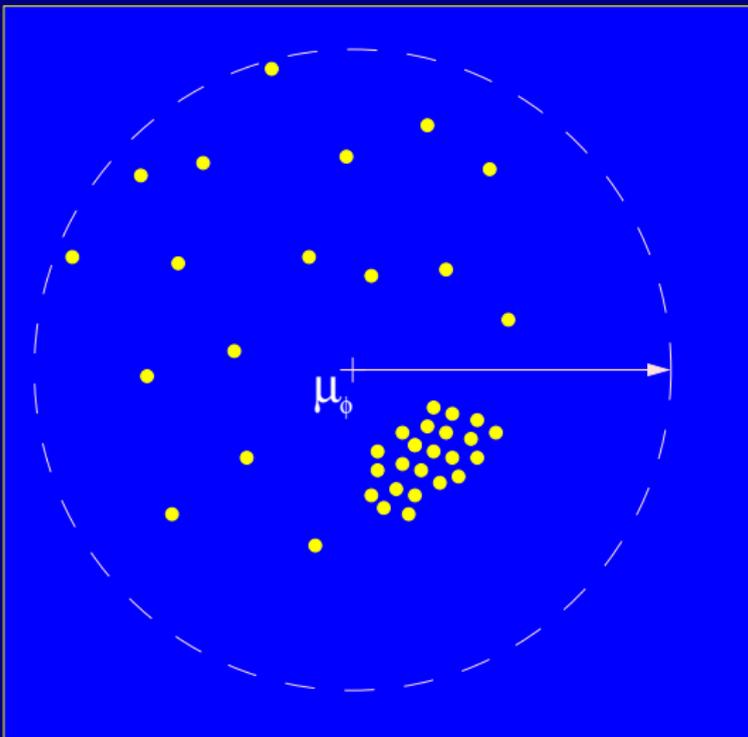
# KDE vs SVDD in feature space



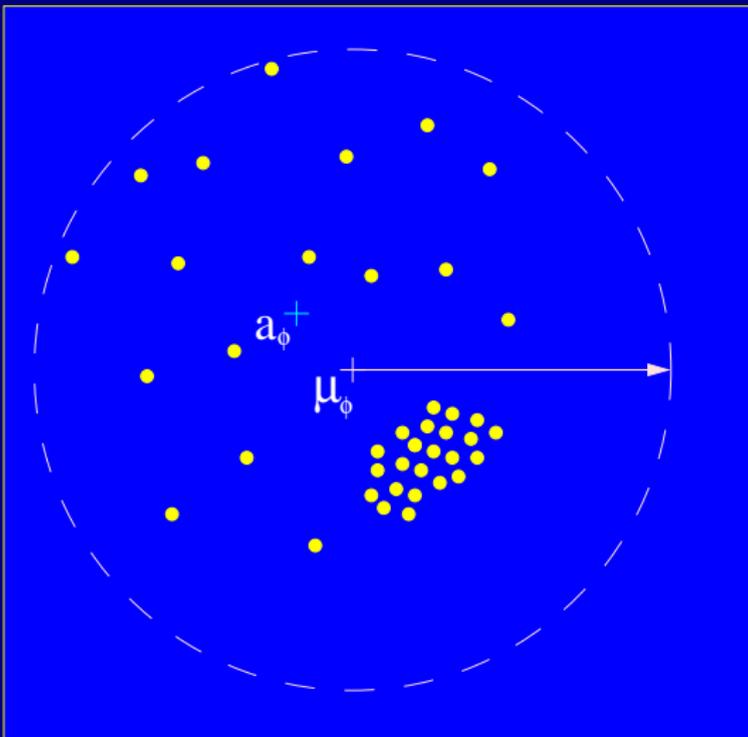
# KDE vs SVDD in feature space



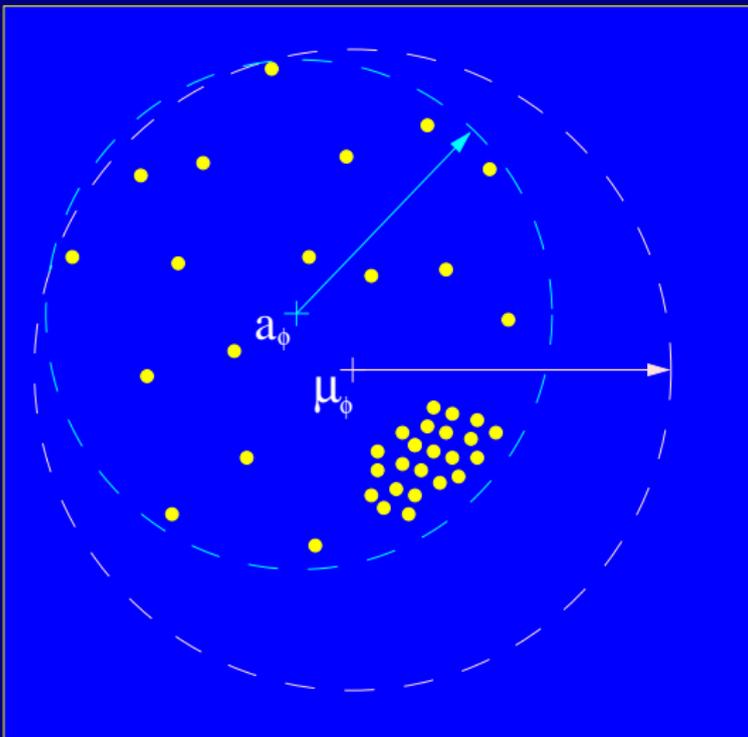
# KDE vs SVDD in feature space



# KDE vs SVDD in feature space



# KDE vs SVDD in feature space



# KDE flattened: project to the data plane

- Given data:  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$
- Map to feature space:  $\{\phi(\mathbf{x}_1), \dots, \phi(\mathbf{x}_N)\}$
- Define centroid:  $\boldsymbol{\mu}_\phi = \frac{1}{N} \sum_{n=1}^N \phi(\mathbf{x}_n)$
- Define:  $\phi_c(\mathbf{x}) = \phi(\mathbf{x}) - \boldsymbol{\mu}_\phi$
- KDE uses distance in feature space:  $\mathcal{A}(\mathbf{x}) = \phi_c(\mathbf{x})^T \phi_c(\mathbf{x})$

# KDE flattened: project to the data plane

- Given data:  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$
- Map to feature space:  $\{\phi(\mathbf{x}_1), \dots, \phi(\mathbf{x}_N)\}$
- Define centroid:  $\boldsymbol{\mu}_\phi = \frac{1}{N} \sum_{n=1}^N \phi(\mathbf{x}_n)$
- Define:  $\phi_c(\mathbf{x}) = \phi(\mathbf{x}) - \boldsymbol{\mu}_\phi$
- KDE uses distance in feature space:  $\mathcal{A}(\mathbf{x}) = \phi_c(\mathbf{x})^T \phi_c(\mathbf{x})$
- Define:  $\boldsymbol{\Phi}_c = [\phi_c(\mathbf{x}_1) \cdots \phi_c(\mathbf{x}_N)]$

# KDE flattened: project to the data plane

- Given data:  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$
- Map to feature space:  $\{\phi(\mathbf{x}_1), \dots, \phi(\mathbf{x}_N)\}$
- Define centroid:  $\boldsymbol{\mu}_\phi = \frac{1}{N} \sum_{n=1}^N \phi(\mathbf{x}_n)$
- Define:  $\phi_c(\mathbf{x}) = \phi(\mathbf{x}) - \boldsymbol{\mu}_\phi$
- KDE uses distance in feature space:  $\mathcal{A}(\mathbf{x}) = \phi_c(\mathbf{x})^T \phi_c(\mathbf{x})$
- Define:  $\boldsymbol{\Phi}_c = [\phi_c(\mathbf{x}_1) \cdots \phi_c(\mathbf{x}_N)] = \underbrace{V_\phi \Lambda^{1/2} W^T}_{\text{SVD}}$

# KDE flattened: project to the data plane

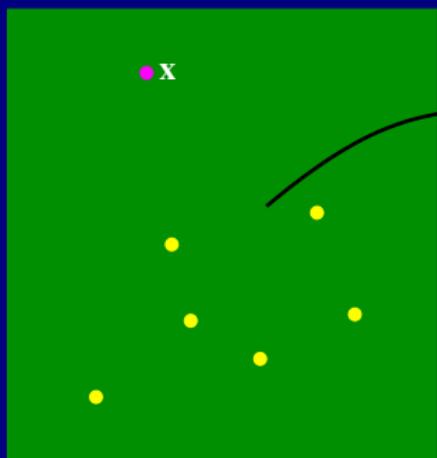
- Given data:  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$
- Map to feature space:  $\{\phi(\mathbf{x}_1), \dots, \phi(\mathbf{x}_N)\}$
- Define centroid:  $\boldsymbol{\mu}_\phi = \frac{1}{N} \sum_{n=1}^N \phi(\mathbf{x}_n)$
- Define:  $\phi_c(\mathbf{x}) = \phi(\mathbf{x}) - \boldsymbol{\mu}_\phi$
- KDE uses distance in feature space:  $\mathcal{A}(\mathbf{x}) = \phi_c(\mathbf{x})^T \phi_c(\mathbf{x})$
- Define:  $\boldsymbol{\Phi}_c = [\phi_c(\mathbf{x}_1) \cdots \phi_c(\mathbf{x}_N)] = \underbrace{V_\phi \Lambda^{1/2} W^T}_{\text{SVD}}$
- Project to data plane:

$$\phi_c^*(\mathbf{x}) = V_\phi^T \phi_c(\mathbf{x}) = \underbrace{\Lambda^{-1/2} W^T \boldsymbol{\Phi}_c^T}_{V_\phi^T} \phi_c(\mathbf{x})$$

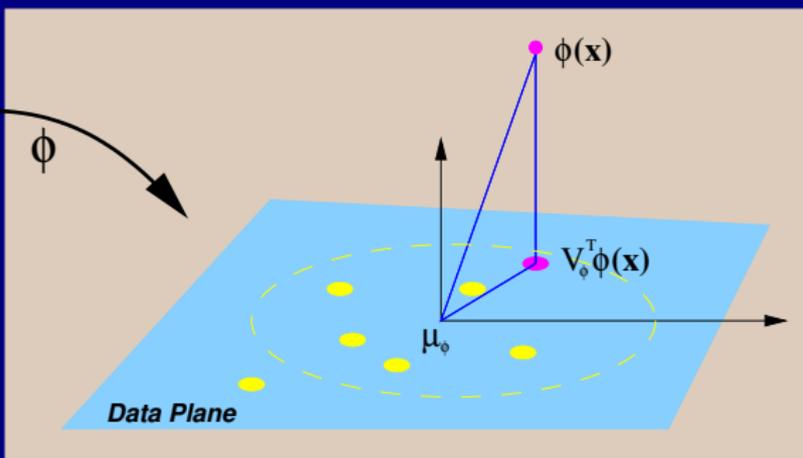
- KDE-flat uses distance after projection to data plane:

$$\mathcal{A}(\mathbf{x}) = \phi_c^*(\mathbf{x})^T \phi_c^*(\mathbf{x}) = \phi_c(\mathbf{x})^T \boldsymbol{\Phi}_c W \Lambda^{-1} W^T \boldsymbol{\Phi}_c^T \phi_c(\mathbf{x})$$

# KDE flattened



**Real space**



**Feature space**

- *Data Plane* is subspace spanned by  $[\phi(\mathbf{x}_1), \dots, \phi(\mathbf{x}_N)]$
- $\phi(\mathbf{x})$  maps potential anomaly  $\mathbf{x}$  to the feature space
- $V_\phi^T \phi(\mathbf{x})$  is projection of  $\phi(\mathbf{x})$  to *Data Plane*
- KDE:  $\mathcal{A}(\mathbf{x}) = \|\phi(\mathbf{x}) - \mu_\phi\|^2$
- KDE-flat:  $\mathcal{A}(\mathbf{x}) = \|V_\phi^T \phi(\mathbf{x}) - \mu_\phi\|^2$

# KDE flattened: project to the data plane

- Recall: KDE-flat defines

$$\mathcal{A}(\mathbf{x}) = \phi_c^*(\mathbf{x})^T \phi_c^*(\mathbf{x}) = \phi_c(\mathbf{x})^T \Phi_c W \Lambda^{-1} W^T \Phi_c^T \phi_c(\mathbf{x})$$

- Let  $\mathbf{k}(\mathbf{x}) = \Phi_c^T \phi(\mathbf{x}) \in \mathbb{R}^N$

$$\mathbf{k}(\mathbf{x}) = \begin{bmatrix} \kappa(\mathbf{x}, \mathbf{x}_1) - (1/N) \sum_{n=1}^N \kappa(\mathbf{x}, \mathbf{x}_n) \\ \vdots \\ \kappa(\mathbf{x}, \mathbf{x}_N) - (1/N) \sum_{n=1}^N \kappa(\mathbf{x}, \mathbf{x}_n) \end{bmatrix}$$

- Let  $\boldsymbol{\mu}_k = \frac{1}{N} \sum_{n=1}^N \mathbf{k}(\mathbf{x}_n)$
- Centered Gram matrix:  $K_c = \Phi_c^T \Phi_c = W \Lambda W^T \in \mathbb{R}^{N \times N}$
- Then, KDE-flat given by

$$\mathcal{A}(\mathbf{x}) = [\mathbf{k}(\mathbf{x}) - \boldsymbol{\mu}_k]^T K_c^{-1} [\mathbf{k}(\mathbf{x}) - \boldsymbol{\mu}_k]$$

# Kernel RX

- Apply kernel trick to RX:  $\mathcal{A}(\mathbf{x}) = \phi_c(\mathbf{x})^T C_\phi^{-1} \phi_c(\mathbf{x})$ , where

$$\begin{aligned} C_\phi &= \sum_{n=1}^N \phi_c(\mathbf{x}_n) \phi_c(\mathbf{x}_n)^T \\ &= \mathbf{\Phi}_c \mathbf{\Phi}_c^T = \underbrace{V_\phi \Lambda^{1/2} W^T}_{\mathbf{\Phi}_c} \underbrace{W \Lambda^{1/2} V_\phi^T}_{\mathbf{\Phi}_c^T} = V_\phi \Lambda V_\phi^T \end{aligned}$$

- $C_\phi$  is *not* invertible; use pseudoinverse:  $C_\phi^{-1} = V_\phi \Lambda^{-1} V_\phi^T$

- Anomalousness:  $\mathcal{A}(\mathbf{x}) = \phi_c(\mathbf{x})^T \underbrace{V_\phi \Lambda^{-1} V_\phi^T}_{C_\phi^{-1}} \phi_c(\mathbf{x})$

- Use  $V_\phi = \mathbf{\Phi}_c W \Lambda^{-1/2}$  to obtain

$$\begin{aligned} \mathcal{A}(\mathbf{x}) &= \phi_c(\mathbf{x})^T \mathbf{\Phi}_c W \Lambda^{-2} W^T \mathbf{\Phi}_c^T \phi_c(\mathbf{x}) \\ &= [\mathbf{k}(\mathbf{x}) - \boldsymbol{\mu}_k]^T K_c^{-2} [\mathbf{k}(\mathbf{x}) - \boldsymbol{\mu}_k] \end{aligned}$$

# Summary of kernel anomaly detectors

## ■ Properties

	centroid	distance	flatten	$\mathcal{A} \sim \kappa$
KDE	$\boldsymbol{\mu}_\phi$	Euclidean	no	linear
SVDD	$\mathbf{a}_\phi$	Euclidean	no	linear
KDE-flat	$\boldsymbol{\mu}_\phi$	Euclidean	yes	quadratic
KRX	$\boldsymbol{\mu}_\phi$	Mahalanobis	yes	quadratic

## ■ Formulas for anomalousness

$$\text{KDE: } \mathcal{A}(\mathbf{x}) = 1 - (1/N) \sum_{n=1}^N \kappa(\mathbf{x}, \mathbf{x}_n)$$

$$\text{SVDD: } \mathcal{A}(\mathbf{x}) = 1 - \sum_n a_n \kappa(\mathbf{x}, \mathbf{x}_n)$$

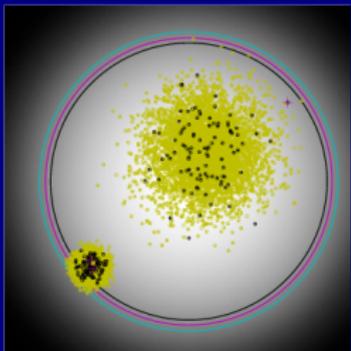
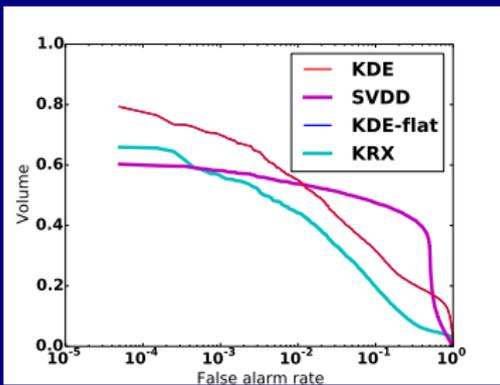
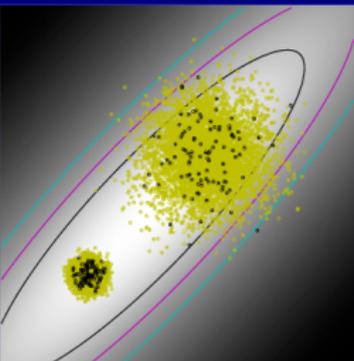
$$\text{KDE-flat: } \mathcal{A}(\mathbf{x}) = [\mathbf{k}(\mathbf{x}) - \boldsymbol{\mu}_k]^T K_c^{-1} [\mathbf{k}(\mathbf{x}) - \boldsymbol{\mu}_k]$$

$$\text{KRX: } \mathcal{A}(\mathbf{x}) = [\mathbf{k}(\mathbf{x}) - \boldsymbol{\mu}_k]^T K_c^{-2} [\mathbf{k}(\mathbf{x}) - \boldsymbol{\mu}_k]$$

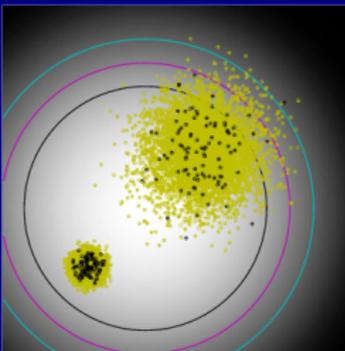
# Numerical experiments - 2D

$$\sigma \rightarrow \infty$$

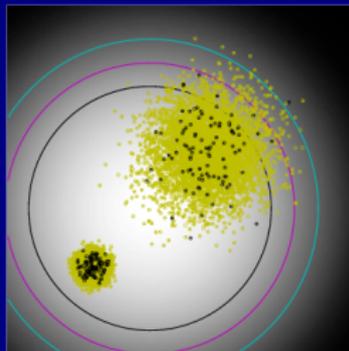
KRX



SVDD



KDE



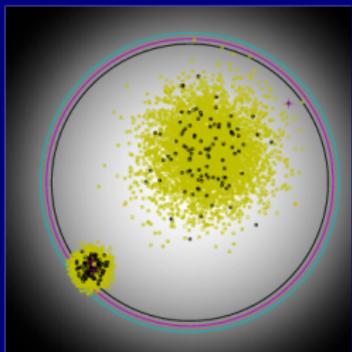
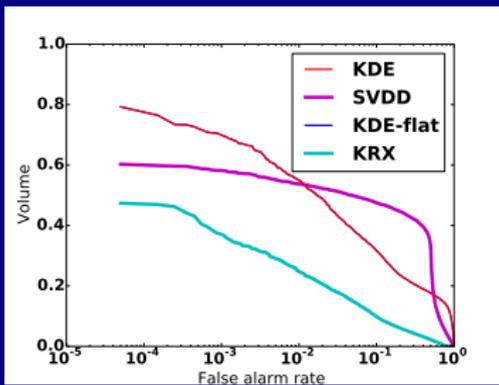
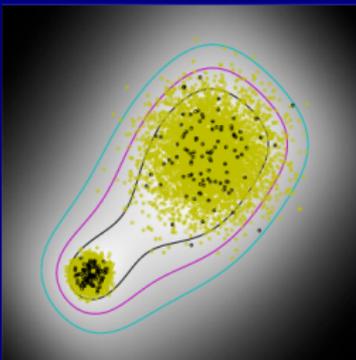
K-flat

false alarm rate contours: 0.05, 0.01, 0.001

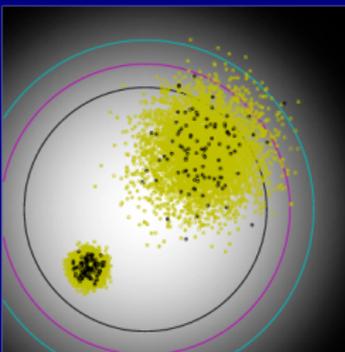
# Numerical experiments - 2D

$$\sigma = 100$$

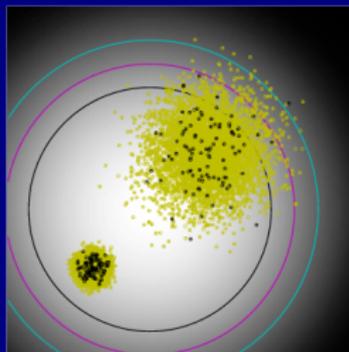
KRX



SVDD



KDE

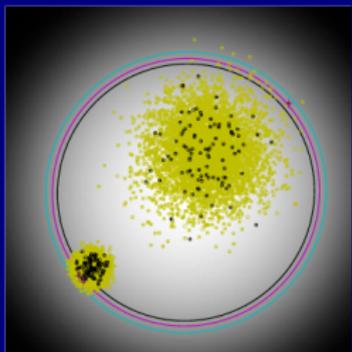
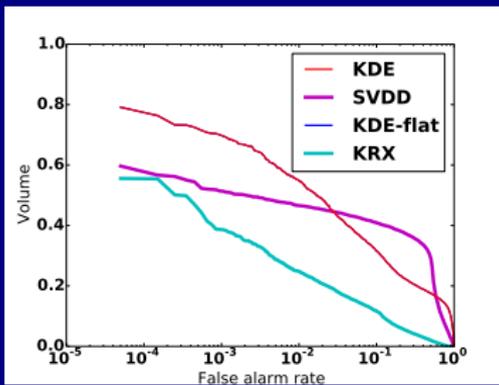
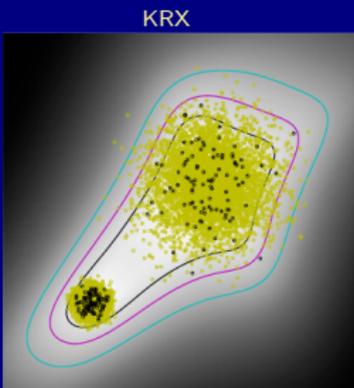


K-flat

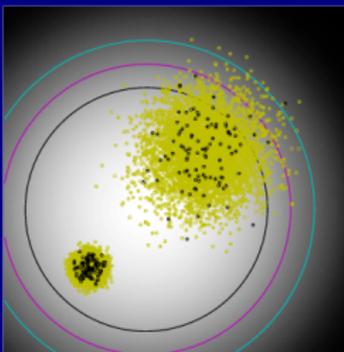
false alarm rate contours: 0.05, 0.01, 0.001

# Numerical experiments - 2D

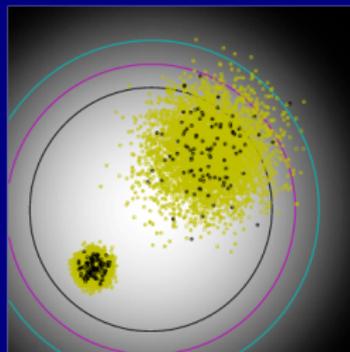
$$\sigma = 30$$



SVDD



KDE



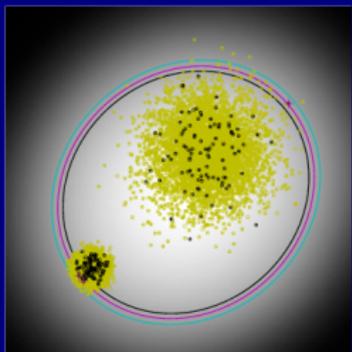
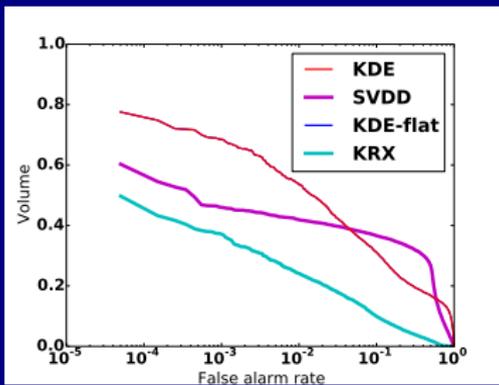
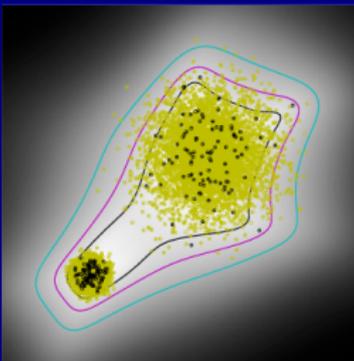
K-flat

false alarm rate contours: 0.05, 0.01, 0.001

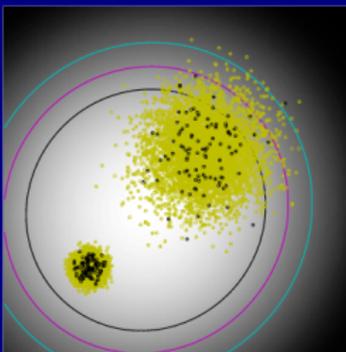
# Numerical experiments - 2D

$$\sigma = 10$$

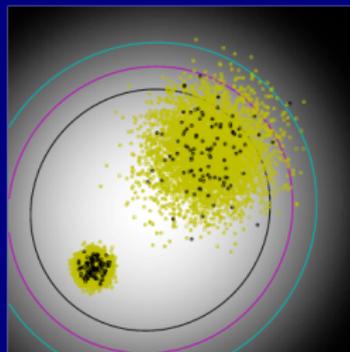
KRX



SVDD



KDE



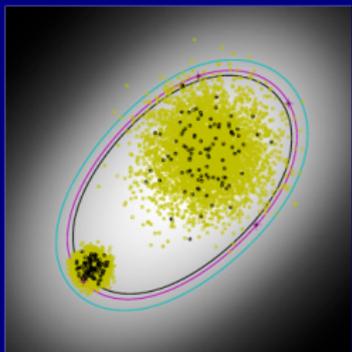
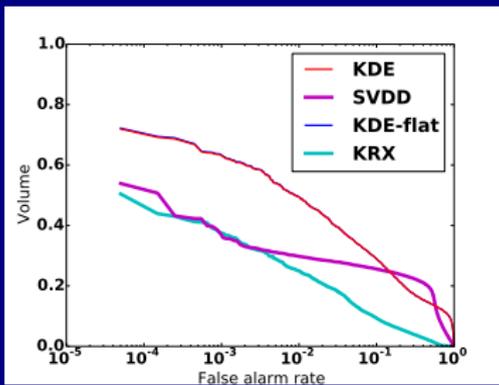
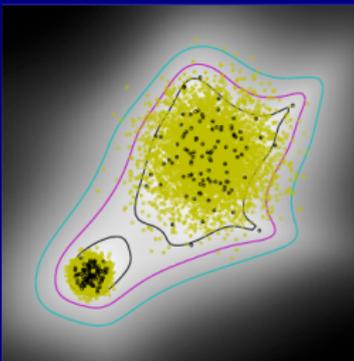
K-flat

false alarm rate contours: 0.05, 0.01, 0.001

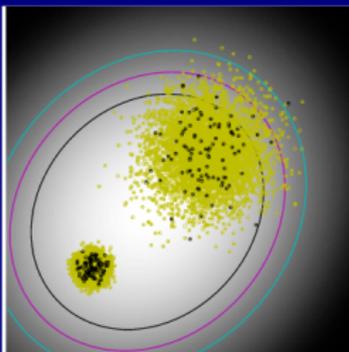
# Numerical experiments - 2D

$$\sigma = 5$$

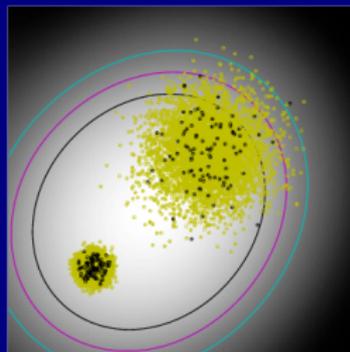
KRX



SVDD



KDE



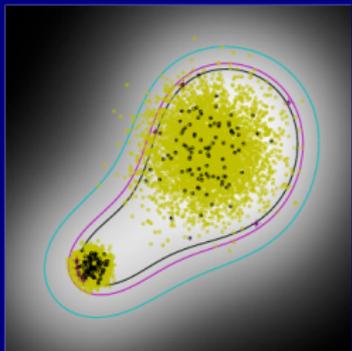
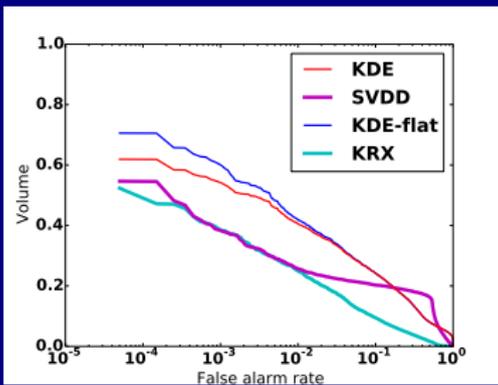
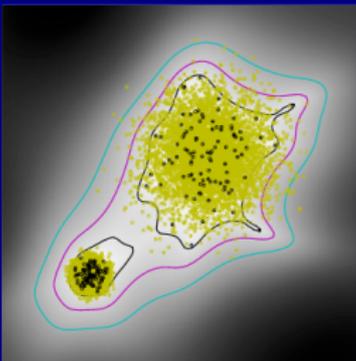
K-flat

false alarm rate contours: 0.05, 0.01, 0.001

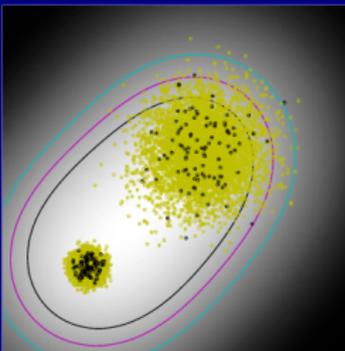
# Numerical experiments - 2D

$$\sigma = 3$$

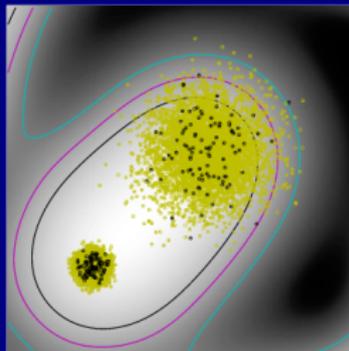
KRX



SVDD



KDE



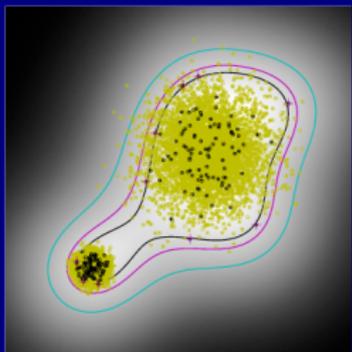
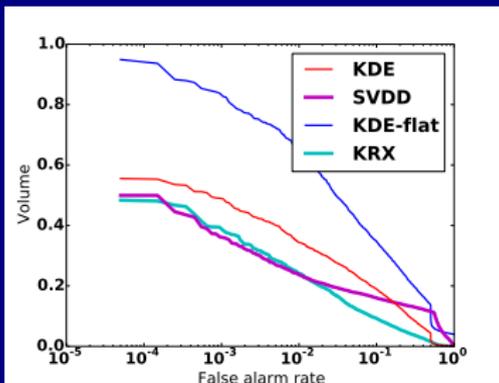
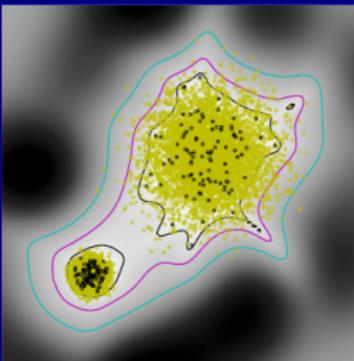
K-flat

false alarm rate contours: 0.05, 0.01, 0.001

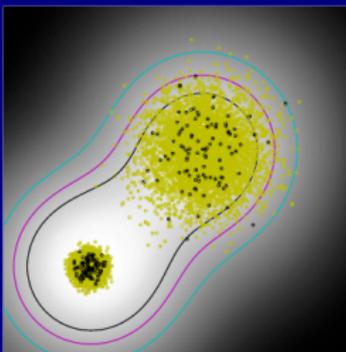
# Numerical experiments - 2D

$$\sigma = 2$$

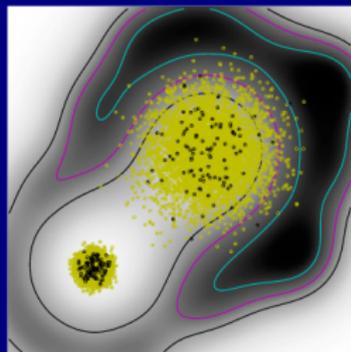
KRX



SVDD



KDE

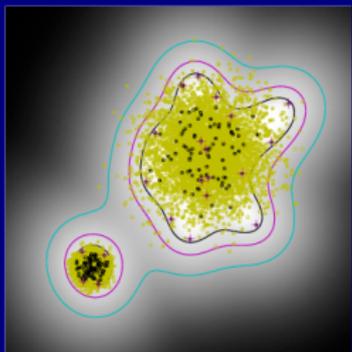
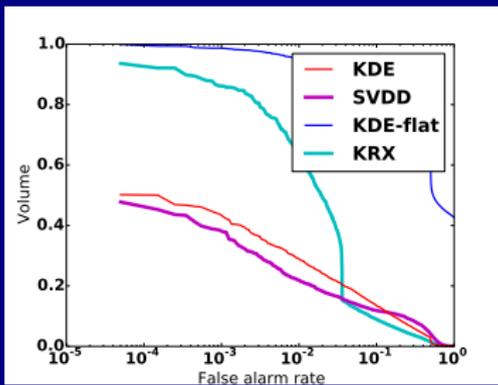
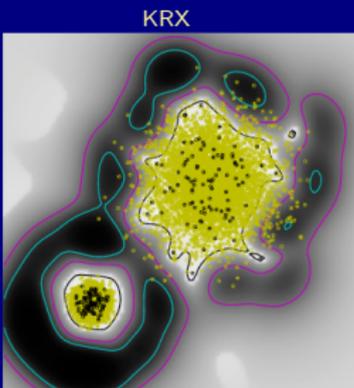


K-flat

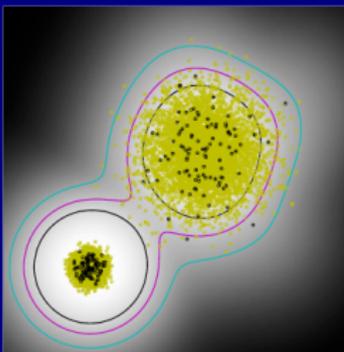
false alarm rate contours: 0.05, 0.01, 0.001

# Numerical experiments - 2D

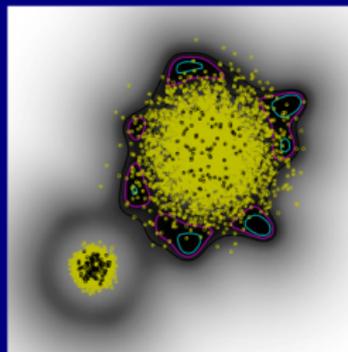
$$\sigma = 1$$



SVDD



KDE



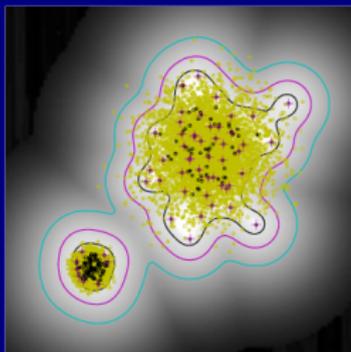
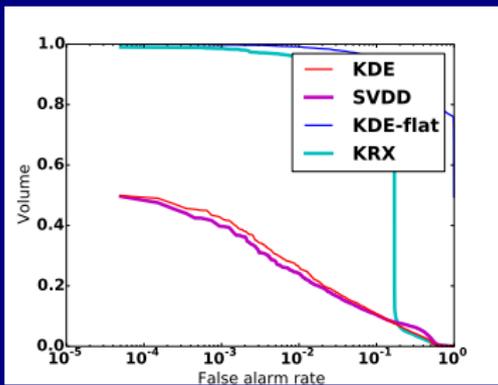
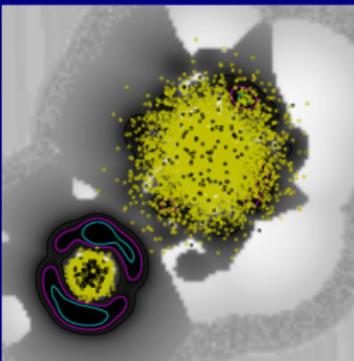
K-flat

false alarm rate contours: 0.05, 0.01, 0.001

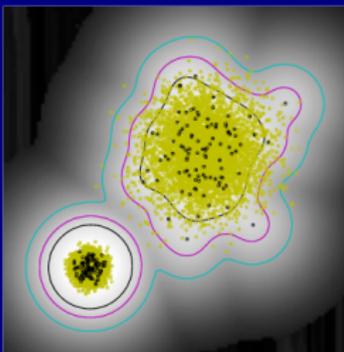
# Numerical experiments - 2D

$$\sigma = 0.5$$

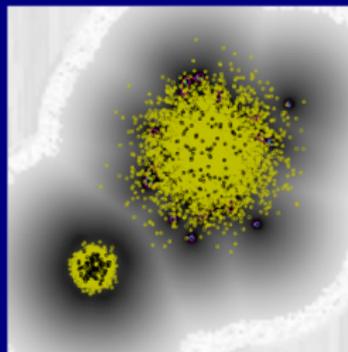
KRX



SVDD



KDE



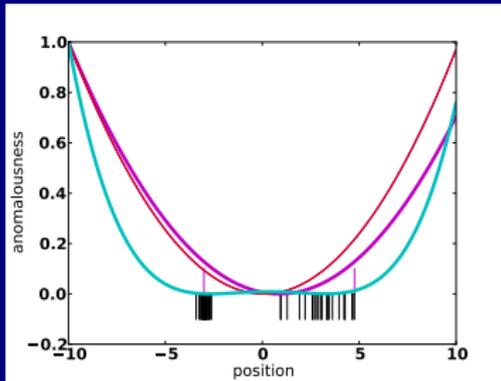
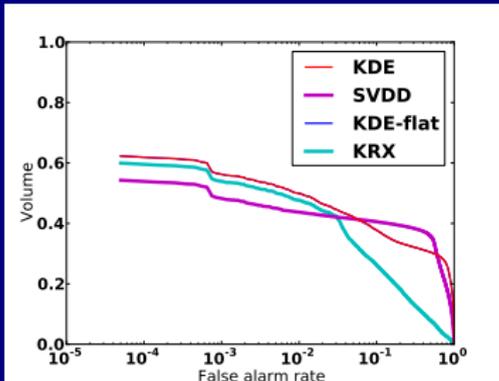
K-flat

false alarm rate contours: 0.05, 0.01, 0.001



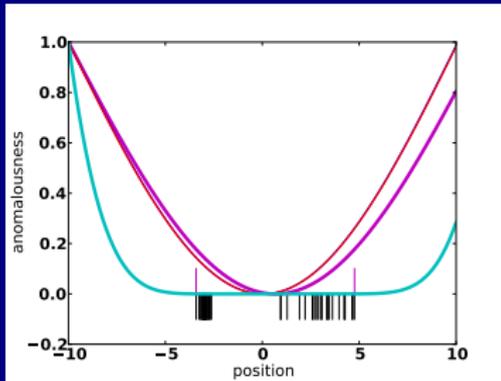
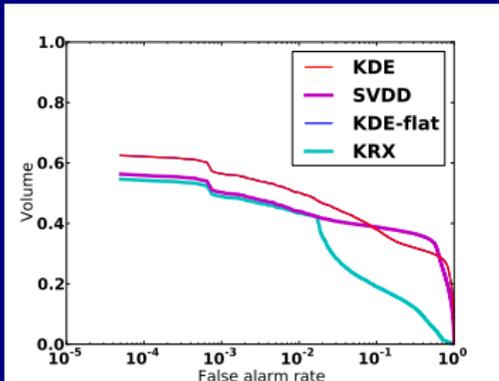
# Numerical experiments - 1D

$$\sigma = 100$$



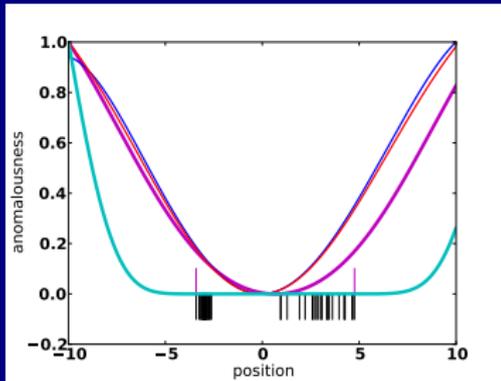
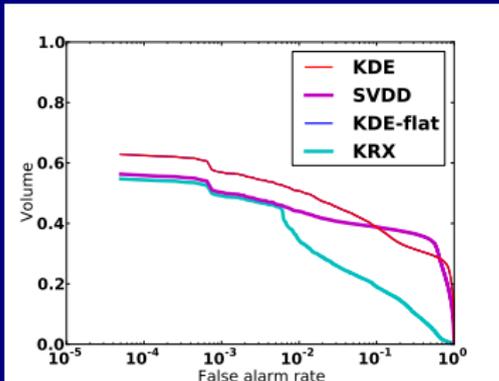
# Numerical experiments - 1D

$$\sigma = 10$$



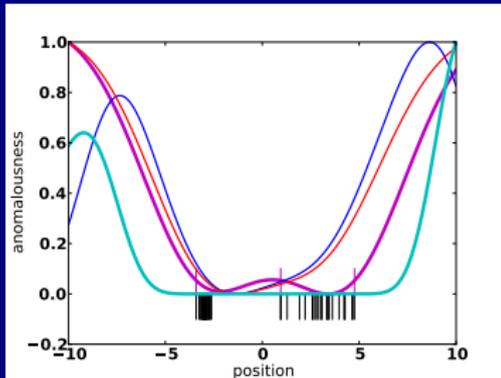
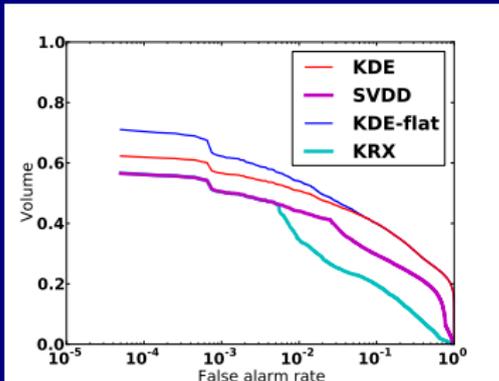
# Numerical experiments - 1D

$$\sigma = 5$$



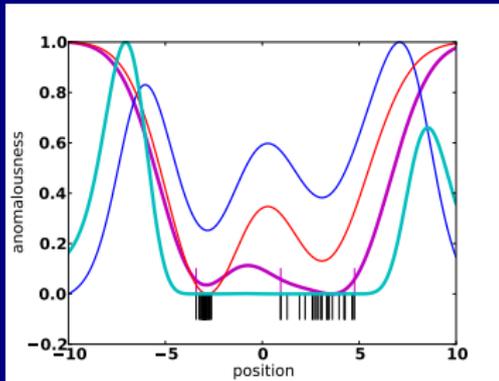
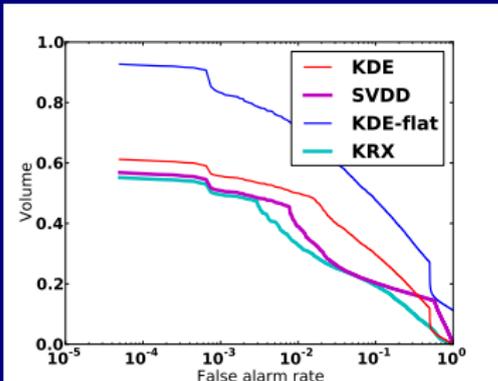
# Numerical experiments - 1D

$$\sigma = 3$$



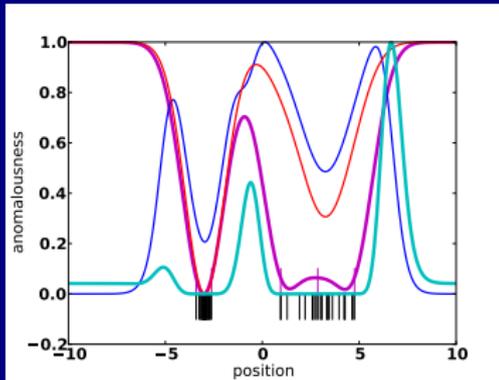
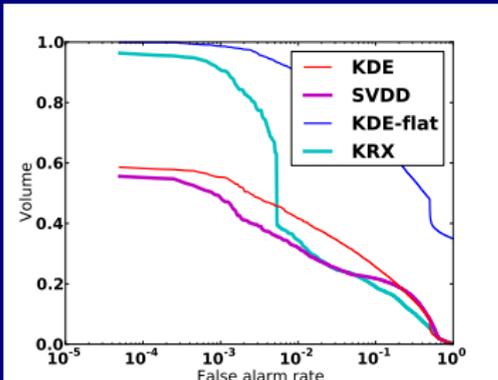
# Numerical experiments - 1D

$$\sigma = 2$$



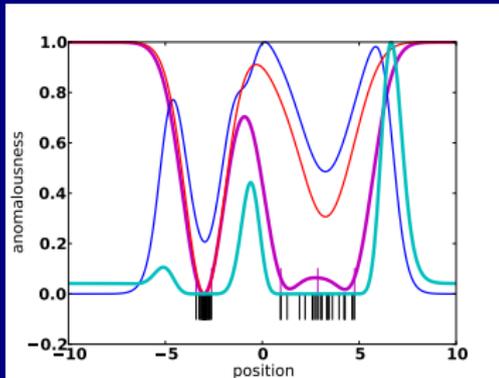
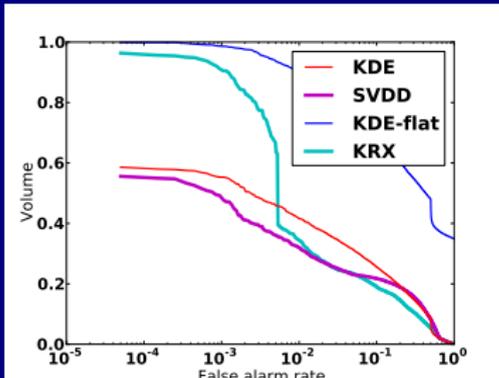
# Numerical experiments - 1D

$$\sigma = 1$$



# Numerical experiments - 1D

$$\sigma = 1$$



- KDE is simple and reasonable, if not optimal
  - Use KDE (not RX) for comparison to new kernel algorithms
- Be wary of projecting to the data plane
  - KDE-flat is a bad idea
  - KRX also projects to the data plane, is that bad?
- For KRX, err on the side of large  $\sigma$ 
  - In  $\sigma \rightarrow \infty$  limit, KRX  $\rightarrow$  RX
  - For small  $\sigma$ , KRX is a disaster!
- SVDD generally more robust to choice of  $\sigma$





Definition

RX

Evaluate

Kernels

K-2d

K-1d

Change



# change

# Anomalous Change Detection (ACD)

“Just because everything is *different* doesn't mean anything has *changed*.”  
–Irene Peter

Ikonos, 23 May 2000

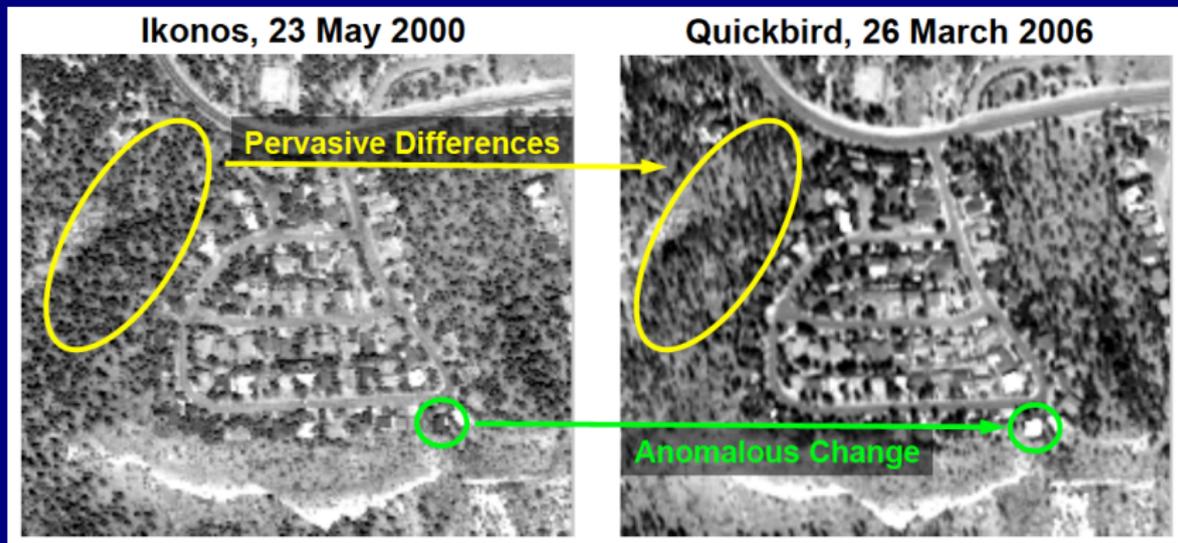


Quickbird, 26 March 2006



# Anomalous Change Detection (ACD)

"Just because everything is *different* doesn't mean anything has *changed*."  
—Irene Peter

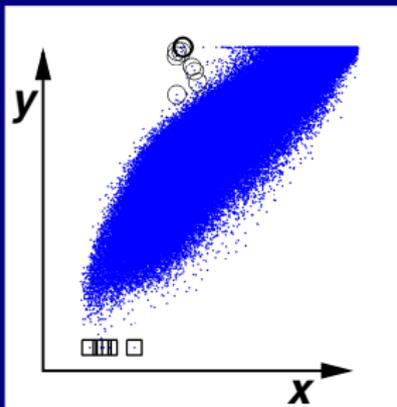


# Pervasive differences vs. Anomalous changes

**x****y**

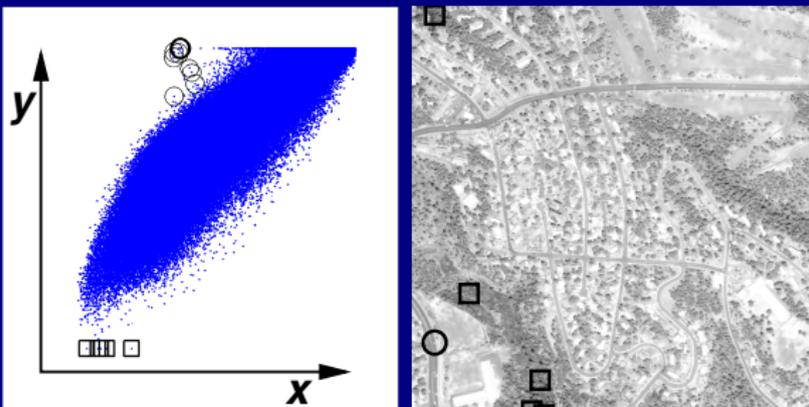
# Anomalies vs. Anomalous changes

- Anomalies: in the tails of  $p(x, y)$
- Anomalous changes:
  - Not unusual components
  - Unusual relationship *between* the components



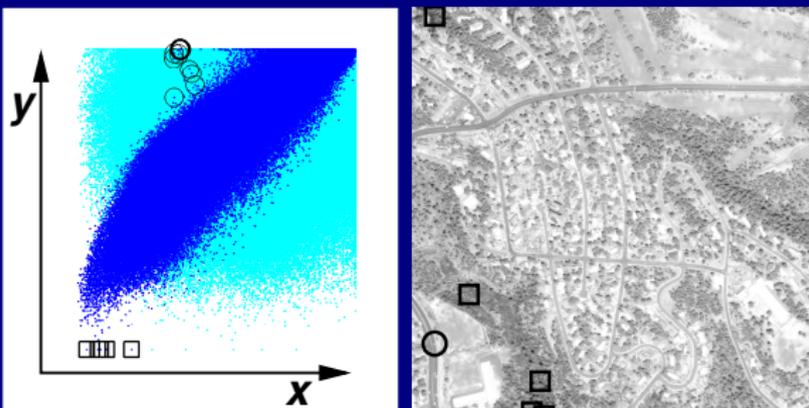
- Circles are anomalous changes
- Squares are anomalies that are not anomalous changes

# Pervasive differences vs. Anomalous changes



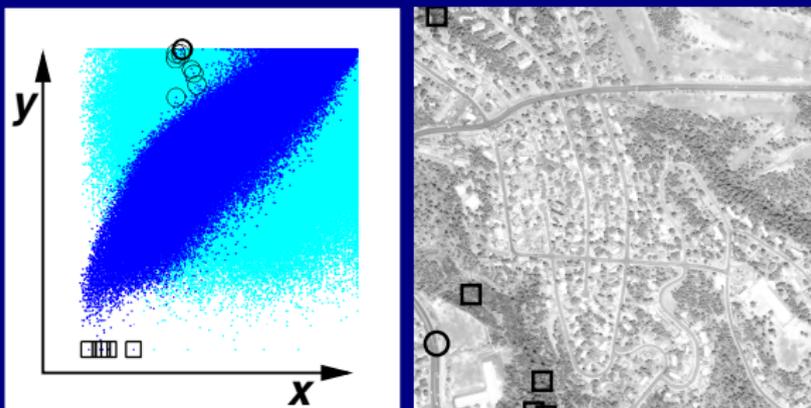
- Pervasive differences:  $p(\mathbf{x}, \mathbf{y})$

# Pervasive differences vs. Anomalous changes



- Pervasive differences:  $p(\mathbf{x}, \mathbf{y})$
- Explicit model for anomalous changes:  $p_a(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{y})$ 
  - Only the change  $\mathbf{x} \rightarrow \mathbf{y}$  is anomalous, not  $\mathbf{x}$  or  $\mathbf{y}$

# Pervasive differences vs. Anomalous changes

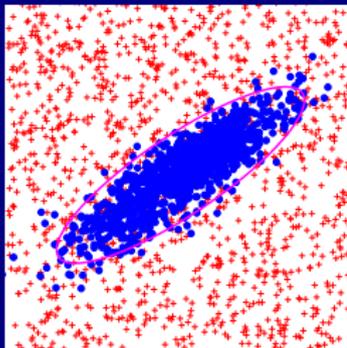


- Pervasive differences:  $p(\mathbf{x}, \mathbf{y})$
- Explicit model for anomalous changes:  $p_a(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{y})$ 
  - Only the change  $\mathbf{x} \rightarrow \mathbf{y}$  is anomalous, not  $\mathbf{x}$  or  $\mathbf{y}$

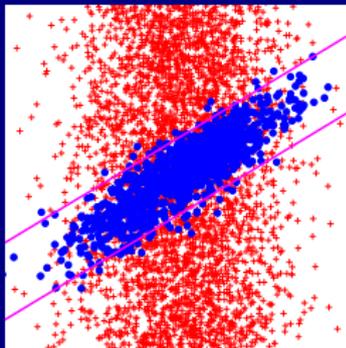
- Likelihood ratio gives optimal classifier:  $\frac{p(\mathbf{x})p(\mathbf{y})}{p(\mathbf{x}, \mathbf{y})}$

# Different anomalous change detectors from different models for anomalous changes

RX



CC



$$\mathcal{A}(\mathbf{x}) = \frac{p_a(\mathbf{x}, \mathbf{y})}{p(\mathbf{x}, \mathbf{y})}$$

RX

$$p_a(\mathbf{x}, \mathbf{y}) = U(\mathbf{x})U(\mathbf{y})$$

CC

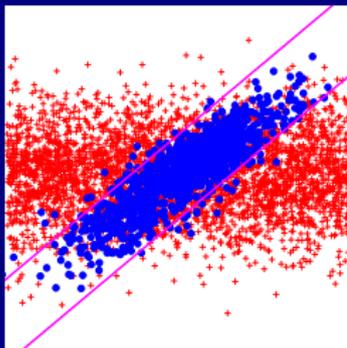
$$p_a(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})U(\mathbf{y})$$

CC

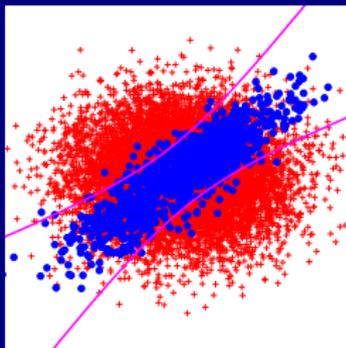
$$p_a(\mathbf{x}, \mathbf{y}) = U(\mathbf{x})p(\mathbf{y})$$

HACD

$$p_a(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{y})$$



CC



HACD

# Obstacles overcome, riddles resolved

Anomalies vs Anomalous changes

$$\begin{array}{l} \mathbf{t} \sim \mathcal{U} \quad \text{vs} \quad \mathbf{t} \sim p_a(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{y}) \\ \mathcal{A}(\mathbf{x}) = 1/p(\mathbf{x}) \quad \text{vs} \quad \mathcal{A}(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{y})/p(\mathbf{x}, \mathbf{y}) \end{array}$$

- Anomalies are rife with conundrums
  - eg,  $p(\mathbf{x})$  depends on coordinates
  - eg,  $1/p(\mathbf{x})$  problematic for subspaces
  - eg, sampling from  $\mathbf{t} \sim \mathcal{U}$  tricky in high dimensions
  
- that are resolved by anomalous changes
  - $p(\mathbf{x})p(\mathbf{y})/p(\mathbf{x}, \mathbf{y})$  invariant to coordinate choice
  - $p(\mathbf{x})p(\mathbf{y})$  has same dimensions as  $p(\mathbf{x}, \mathbf{y})$
  - sampling from  $\mathbf{t} \sim p(\mathbf{x})p(\mathbf{y})$  just resamples data

# Simulation framework for ACD



Base image  
to begin with



Pervasive Differences  
applied to all pixels



Anomalous Change  
applied to one pixel



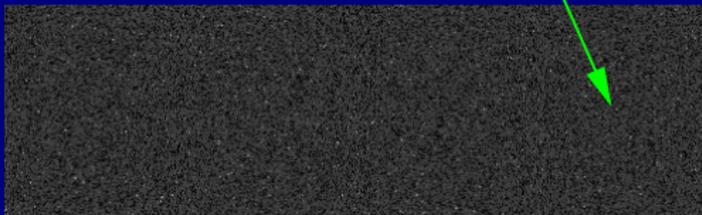
# Simulation framework for ACD



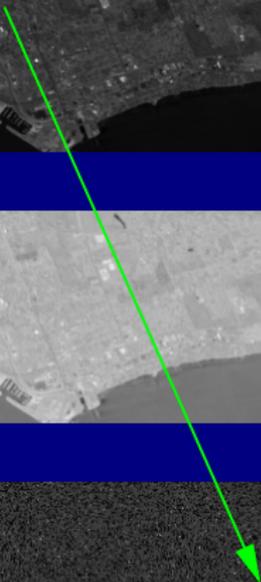
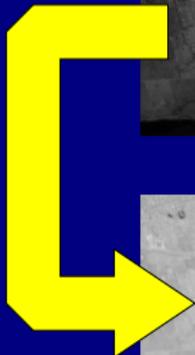
Base image  
to begin with



Pervasive Differences  
applied to all pixels

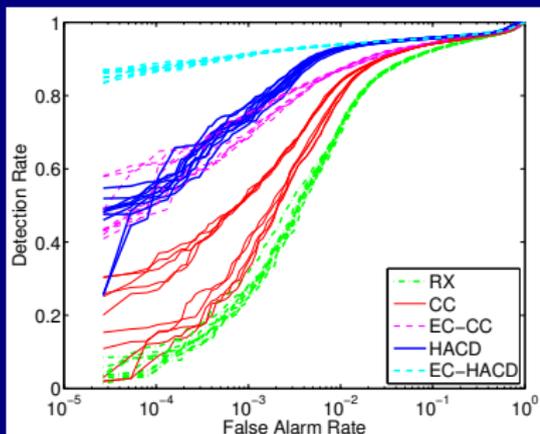


Anomalous Change  
applied to all pixels

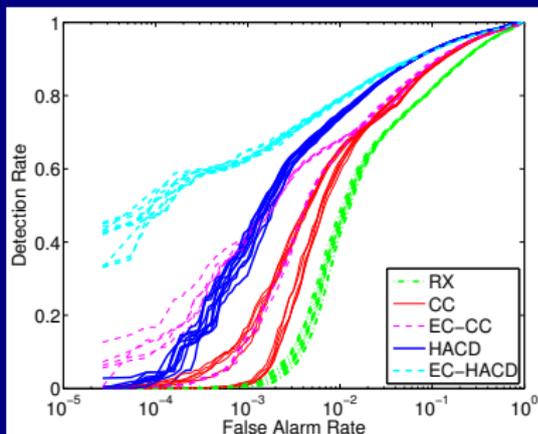


# ROC curves for simulated anomalous changes

split channels



smooth misregistration



- Ten trials: different in-sample/out-of-sample partitions
- CC = ChronoChrome
- HACD = Hyperbolic ACD
- EC = Elliptically Contoured

# AFRL data (Eismann, Meola)

taken Aug 25, 2005



taken Oct 14, 2005

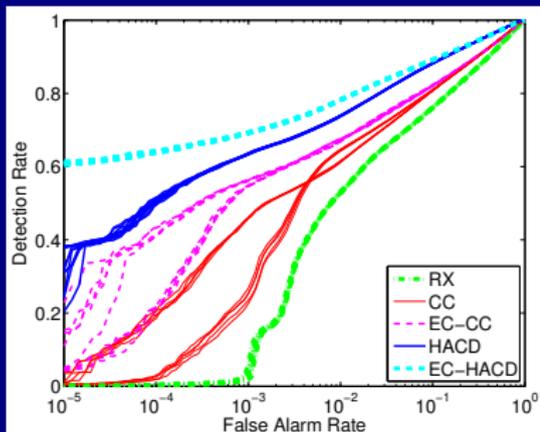


taken Oct 14, 2005, after placing two dark tarps on the grass

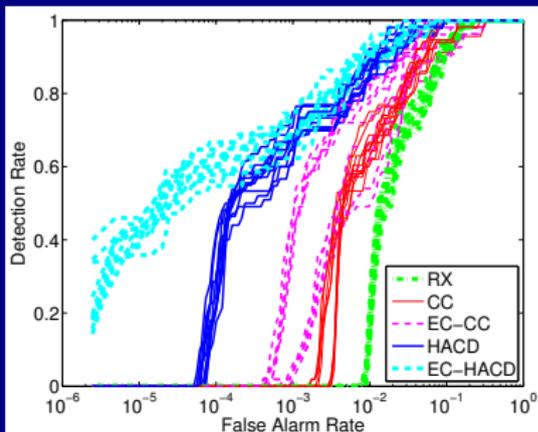


# AFRL data: ROC curves

simulated anomalies

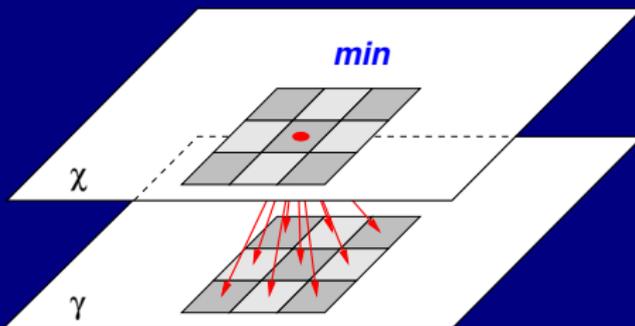


real anomalies



- Ten trials: different in-sample/out-of-sample partitions
- CC = ChronoChrome
- HACD = Hyperbolic ACD
- EC = Elliptically Contoured

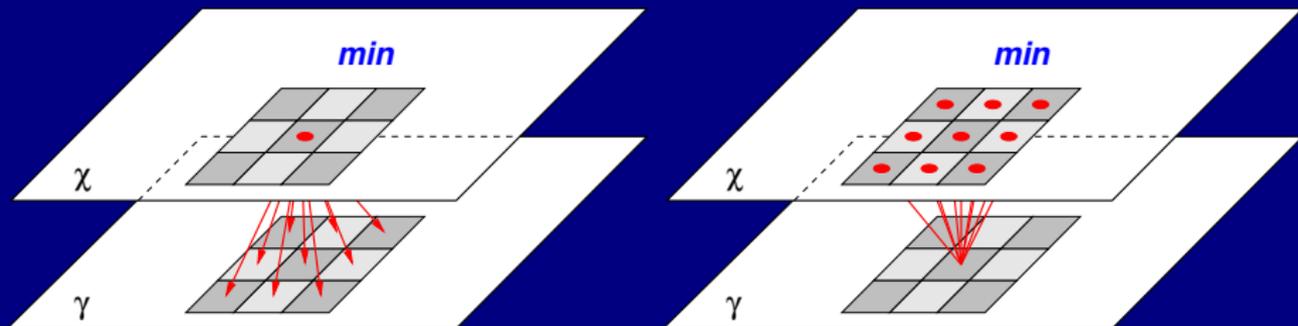
# Misregistration compensation



- LCRA: Local Co-Registration Adjustment

$$A_{k,l} = \min_{m,n} \mathcal{A}(\chi_{k,l}, \gamma_{k+m,l+n})$$

# Misregistration compensation

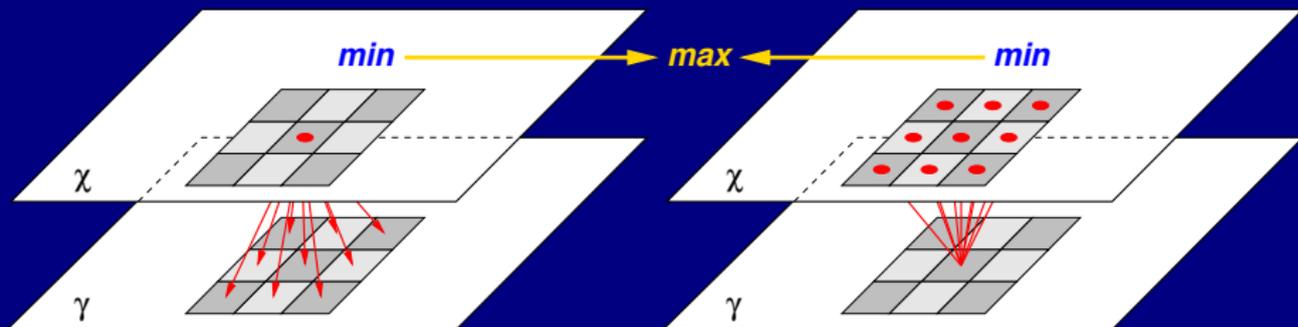


- LCRA: Local Co-Registration Adjustment

$$A_{k,l} = \min_{m,n} \mathcal{A}(\chi_{k,l}, \gamma_{k+m,l+n})$$

- SLCRA: Symmetric LCRA

# Misregistration compensation

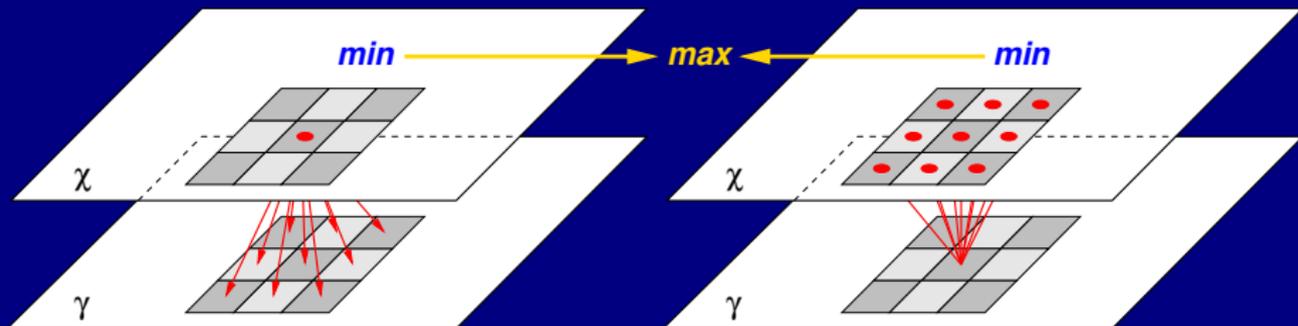


- LCRA: Local Co-Registration Adjustment

$$A_{k,l} = \min_{m,n} \mathcal{A}(\chi_{k,l}, \gamma_{k+m,l+n})$$

- SLCRA: Symmetric LCRA (max of min's)

# Misregistration compensation



- LCRA: Local Co-Registration Adjustment

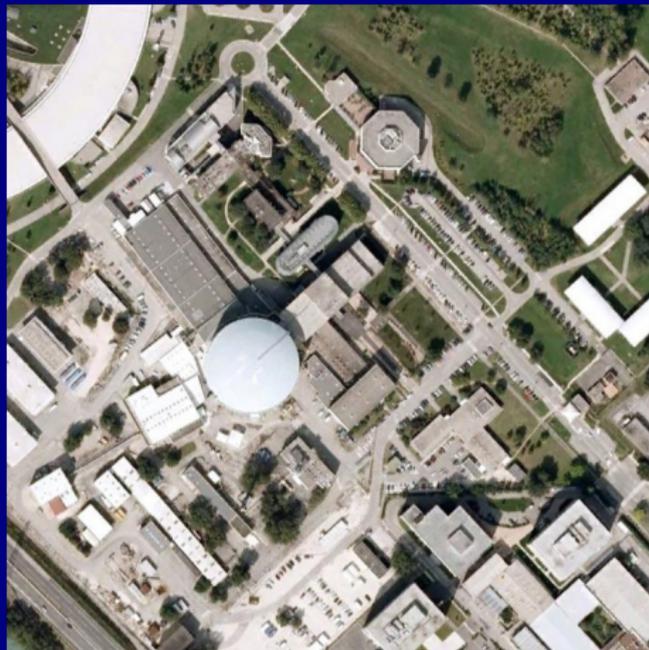
$$A_{k,l} = \min_{m,n} \mathcal{A}(\chi_{k,l}, \gamma_{k+m,l+n})$$

- SLCRA: Symmetric LCRA (max of min's)
- Sub-pixel adjustments
- Covariance re-estimation
- etc.*

# Misregistration compensation

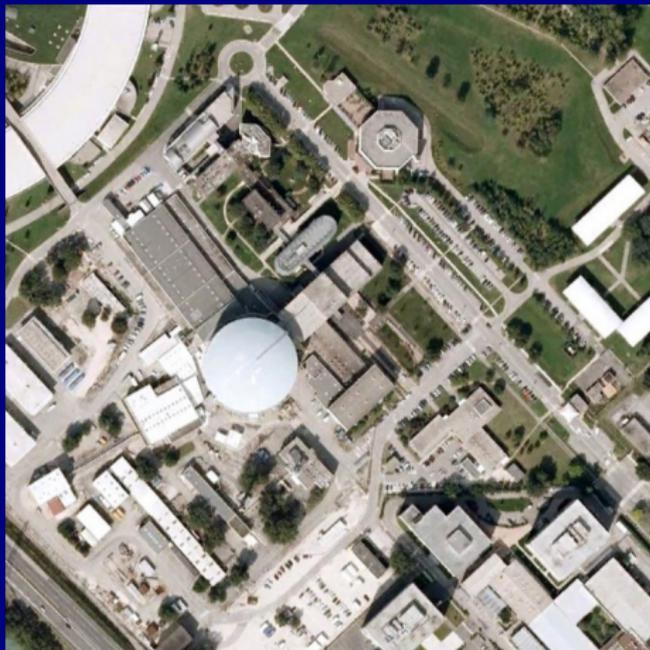


2006

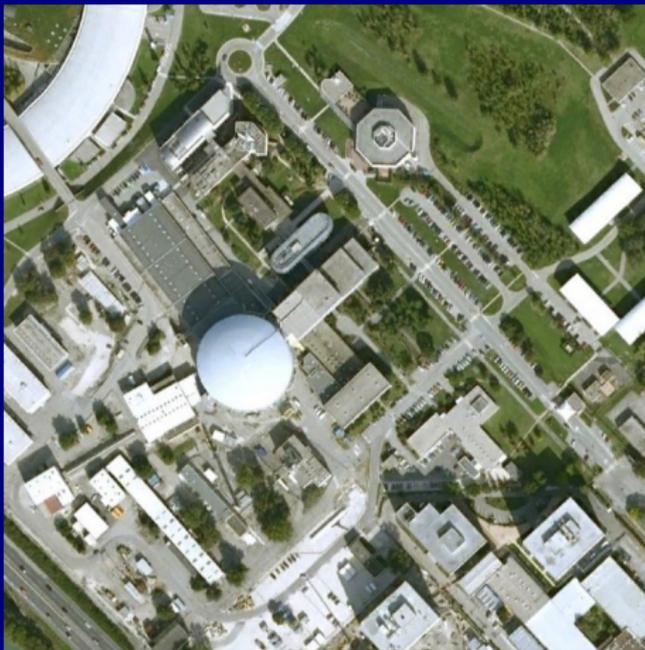


2008

# Misregistration compensation

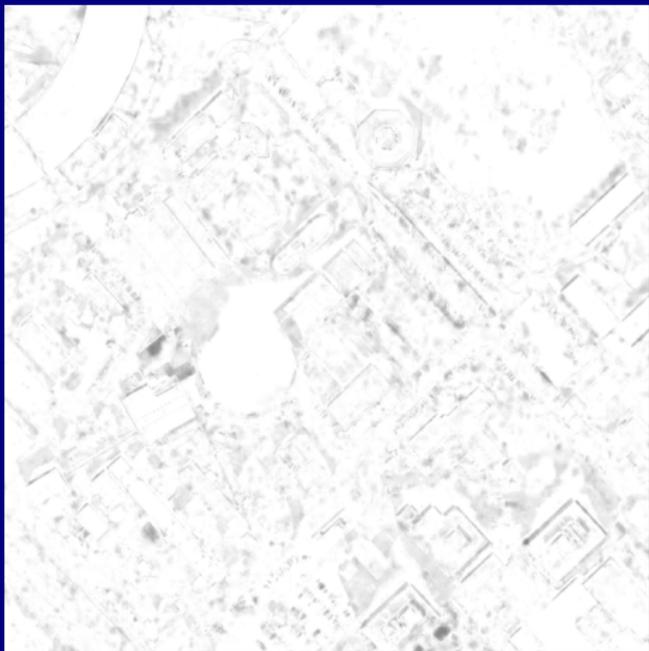


2008

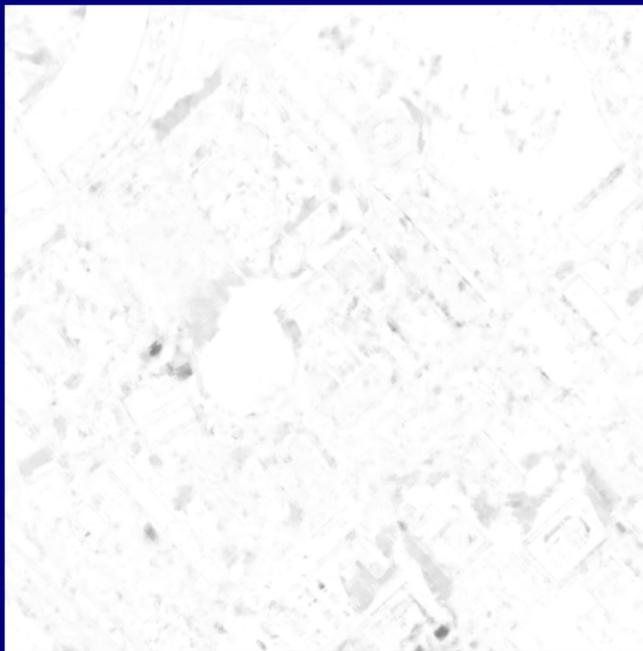


2006

# Misregistration compensation



EC-HACD



EC-HACD + SLCRA





# Conclusion

If we believe that anomaly (and/or anomalous change) detection is interesting, useful, or important;

# Conclusion

If we believe that anomaly (and/or anomalous change) detection is interesting, useful, or important; Then we need to:

1. Employ objective and reliable measures of performance
  - eg, scrambled pixels for anomalous *change* detection
  - eg, volume vs false alarm rate, for anomaly detection
  - challenge: high dimensions
  - challenge: subspaces and coordinate choices
  - challenge: non-probabilistic approaches
    - ▶ eg, kernels, graphs, manifolds, sparse models, etc.
2. Define more explicitly what we mean by “anomaly”
  - a·n·o·m·a·l·y: target defined by a probability distribution that in general is broad and flat, and in particular instances can be specified precisely; eg,  $\mathbf{t} \sim \mathcal{U}$ .

# Conclusion

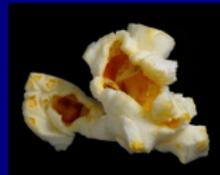
If we believe that anomaly (and/or anomalous change) detection is interesting, useful, or important; Then we need to:

1. Employ objective and reliable measures of performance
  - eg, scrambled pixels for anomalous *change* detection
  - eg, volume vs false alarm rate, for anomaly detection
  - challenge: high dimensions
  - challenge: subspaces and coordinate choices
  - challenge: non-probabilistic approaches
    - ▶ eg, kernels, graphs, manifolds, sparse models, etc.
2. Define more explicitly what we mean by “anomaly”
  - a·n·o·m·a·l·y: target defined by a probability distribution that in general is broad and flat, and in particular instances can be specified precisely; eg,  $\mathbf{t} \sim \mathcal{U}$ .

# Conclusion

If we believe that anomaly (and/or anomalous change) detection is interesting, useful, or important; Then we need to:

1. Employ objective and reliable measures of performance
  - eg, scrambled pixels for anomalous *change* detection
  - eg, volume vs false alarm rate, for anomaly detection
  - challenge: high dimensions
  - challenge: subspaces and coordinate choices
  - challenge: non-probabilistic approaches
    - ▶ eg, kernels, graphs, manifolds, sparse models, etc.
2. Define more explicitly what we mean by “anomaly”
  - a·nom·a·ly: target defined by a probability distribution that in general is broad and flat, and in particular instances can be specified precisely; eg,  $\mathbf{t} \sim \mathcal{U}$ .



No, really, what *are* you hungry for?