

Sequential Feature Explanations for Anomaly Detection

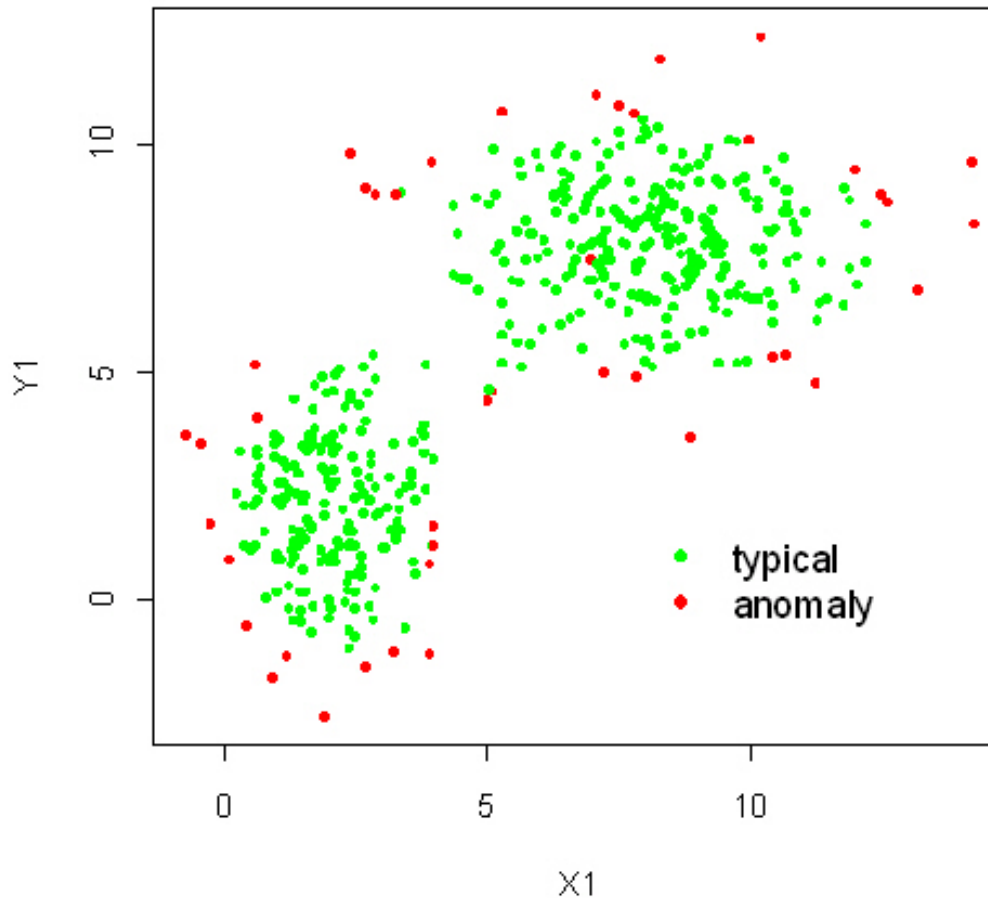
Md Amran Siddiqui, Alan Fern, Thomas G. Dietterich and
Weng-Keen Wong

School of EECS
Oregon State University

Anomaly Detection

Anomalies : points that are generated by a process that is distinct from the process generating “normal” points

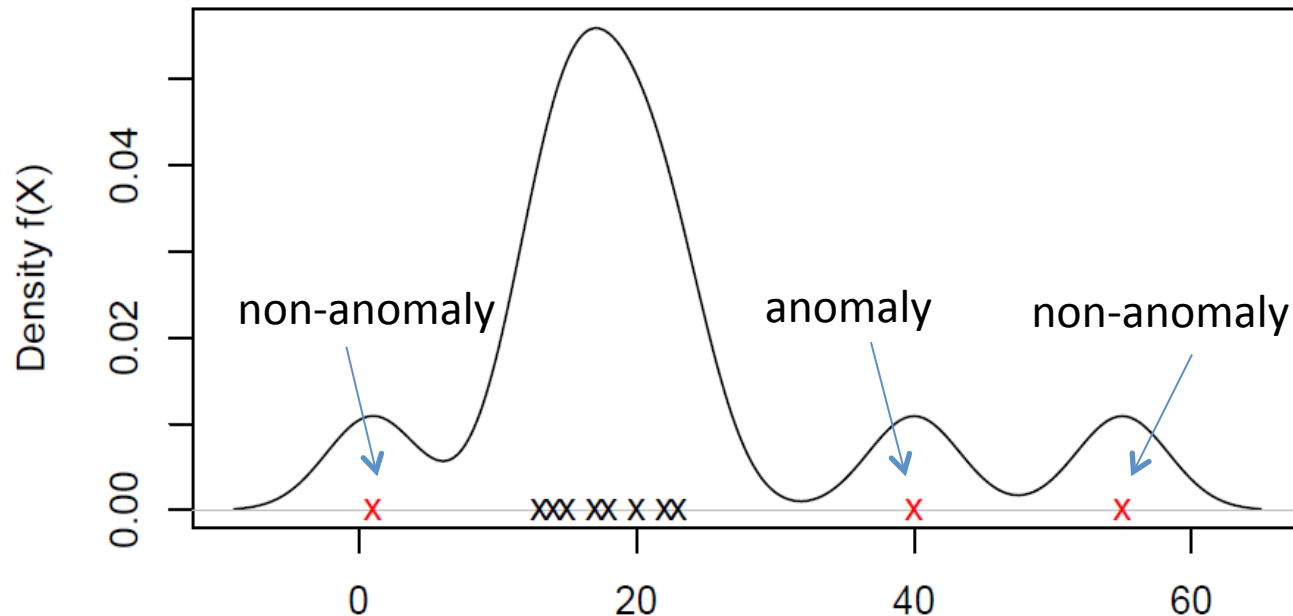
In this talk **Anomaly = Threat**



Anomaly Detectors

We focus on **density-based anomaly detectors**

Statistical Outliers : points with low density values



Not all statistical outliers are anomalies of interest
(statistics versus semantics)

Anomaly Detection Pipeline

Data Points

Threats &
Non-Threats

Anomaly Detection Pipeline

Data Points

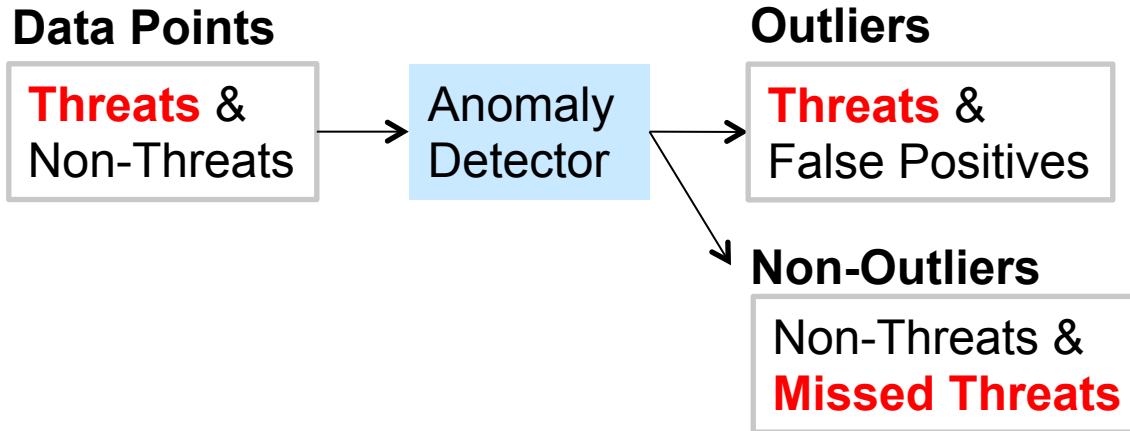
Threats &
Non-Threats



```
graph LR; A[Data Points: Threats & Non-Threats] --> B[Anomaly Detector]
```

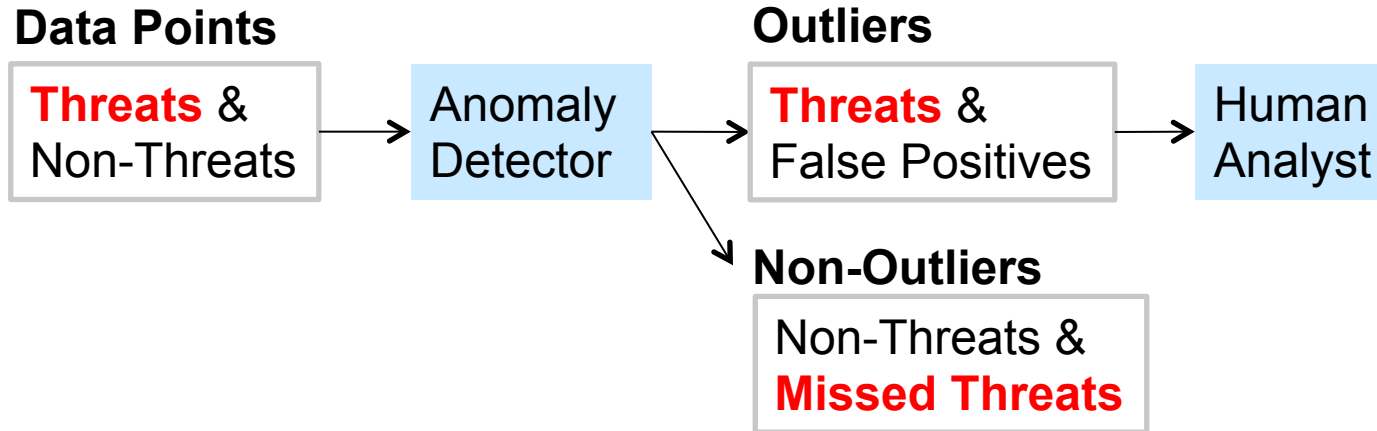
Anomaly
Detector

Anomaly Detection Pipeline



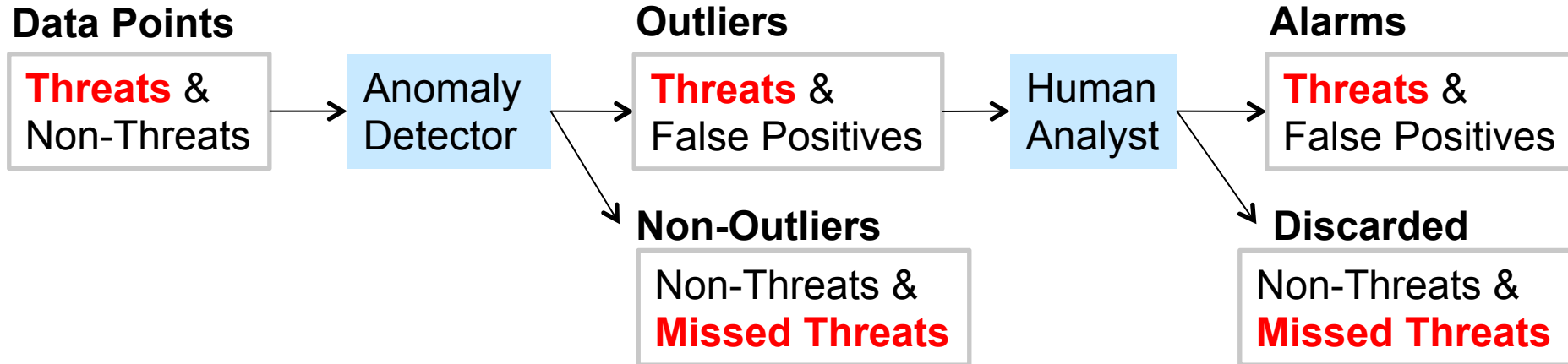
- Type 1 Missed Threats = Anomaly Detector's False Negatives
 - Reduce by improving anomaly detector

Anomaly Detection Pipeline



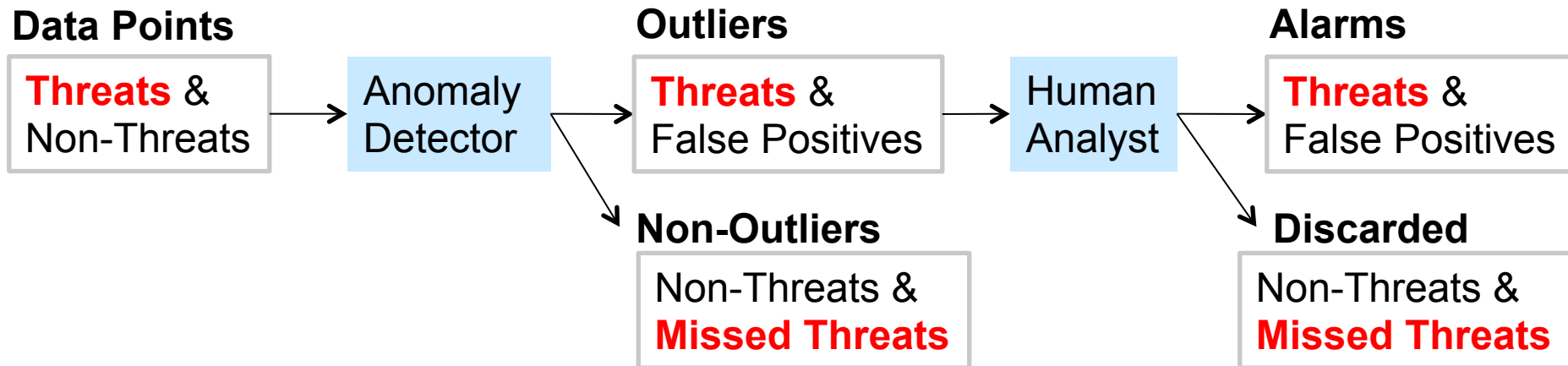
- Type 1 Missed Threats = Anomaly Detector's False Negatives
 - Reduce by improving anomaly detector

Anomaly Detection Pipeline



- Type 1 Missed Threats = Anomaly Detector's False Negatives
 - Reduce by improving anomaly detector

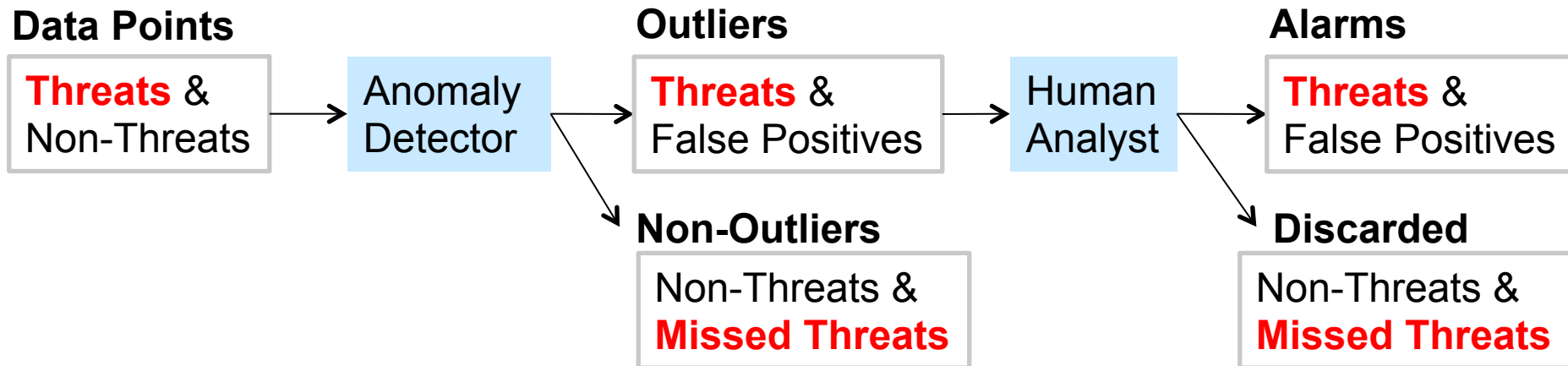
Anomaly Detection Pipeline



- **Type 1 Missed Threats = Anomaly Detector's False Negatives**
 - Reduce by improving anomaly detector
- **Type 2 Missed Threats = Analyst's False Negatives**
 - Can occur due to information overload and time constraints

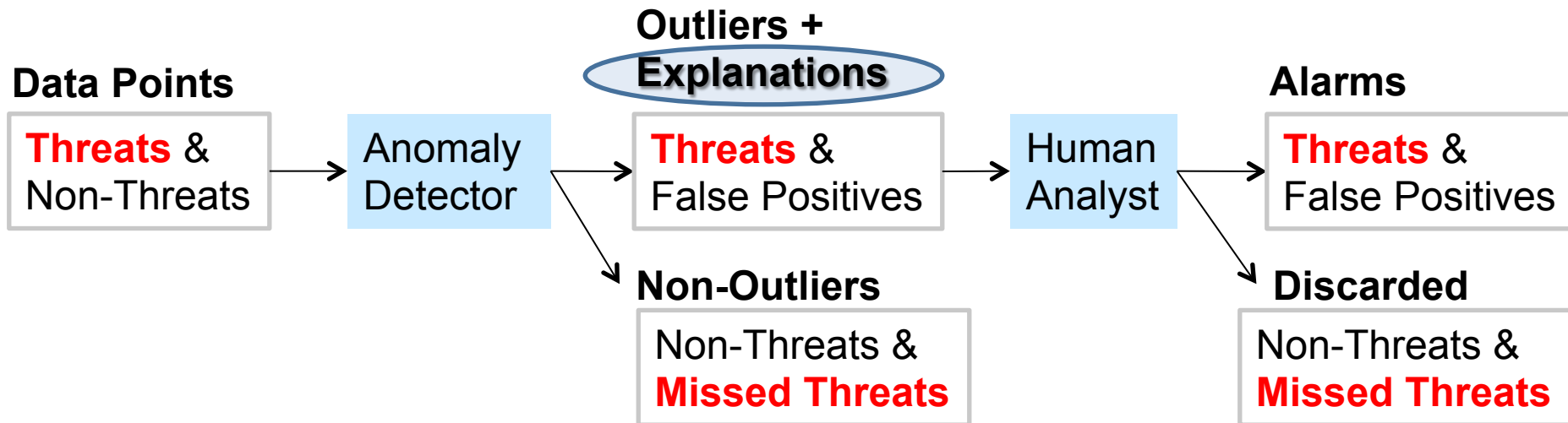
How can we reduce type 2 errors?

Anomaly Detection Pipeline



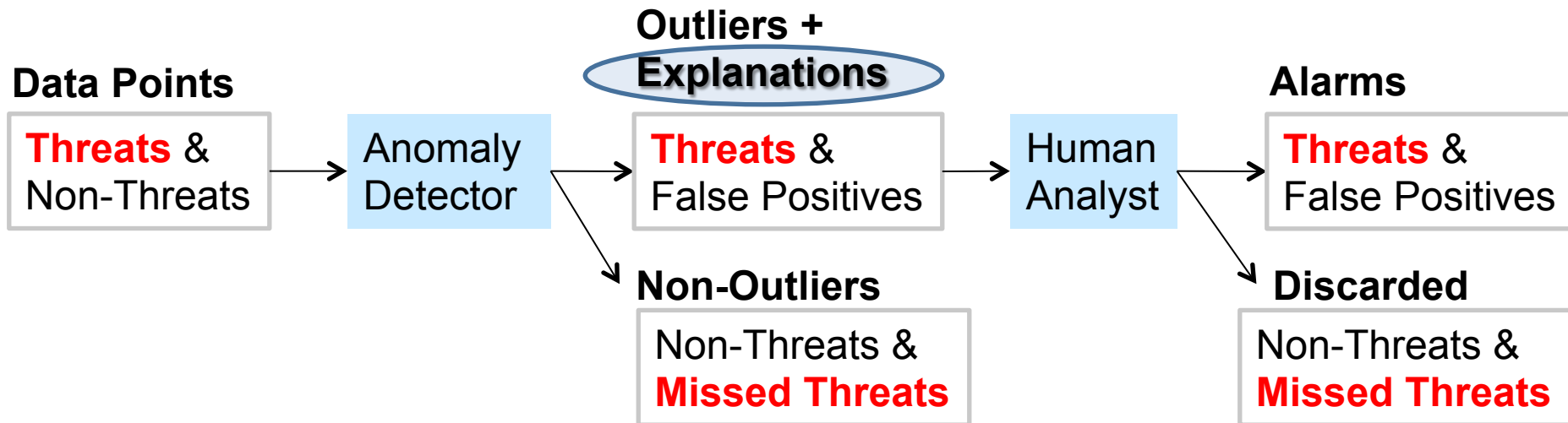
- **Goal:** reduce analyst effort for correctly detecting outliers that are threats

Anomaly Detection Pipeline



- **Goal:** reduce analyst effort for correctly detecting outliers that are threats
- **How:** provide analyst with “explanations” of outlier points

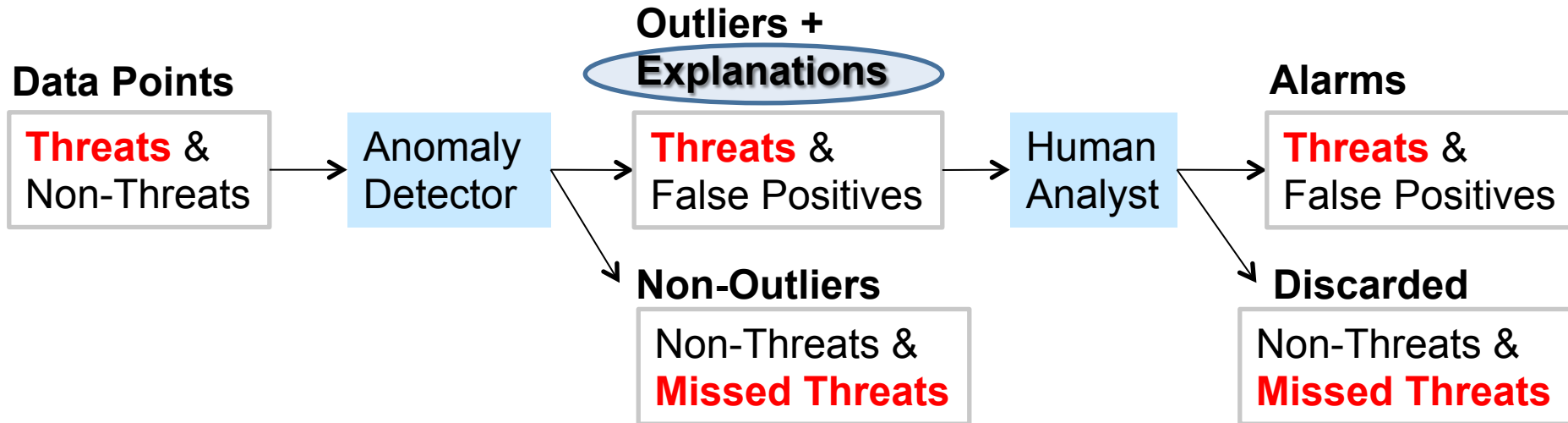
Anomaly Detection Pipeline



- **Goal:** reduce analyst effort for correctly detecting outliers that are threats
- **How:** provide analyst with “explanations” of outlier points

Why did the detector consider an object to be an outlier?

Anomaly Detection Pipeline

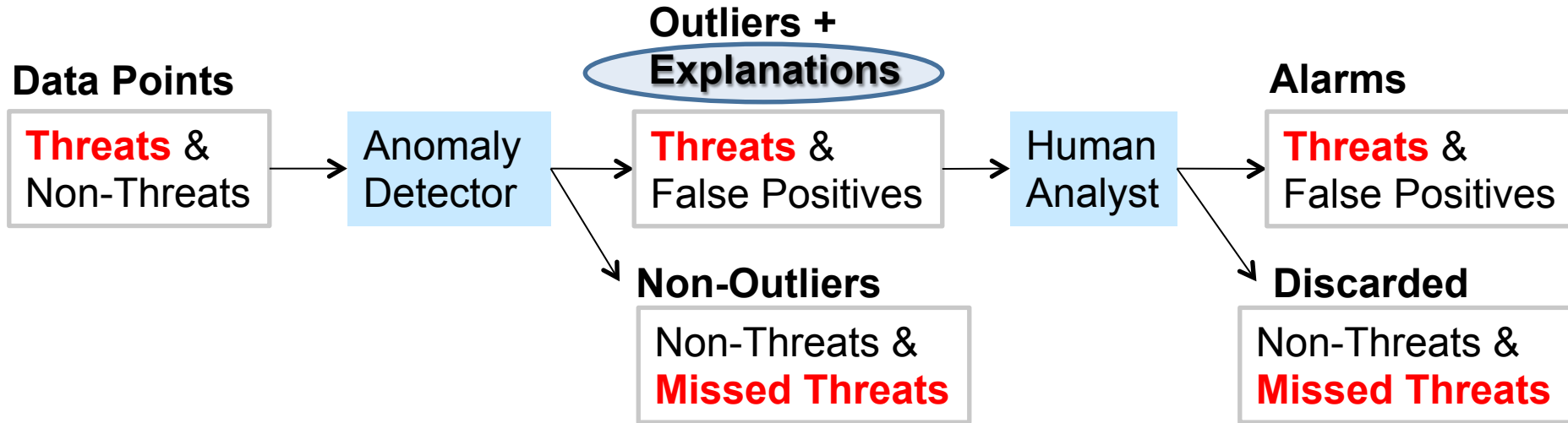


- **Goal:** reduce analyst effort for correctly detecting outliers that are threats
- **How:** provide analyst with “explanations” of outlier points

Why did the detector consider an object to be an outlier?

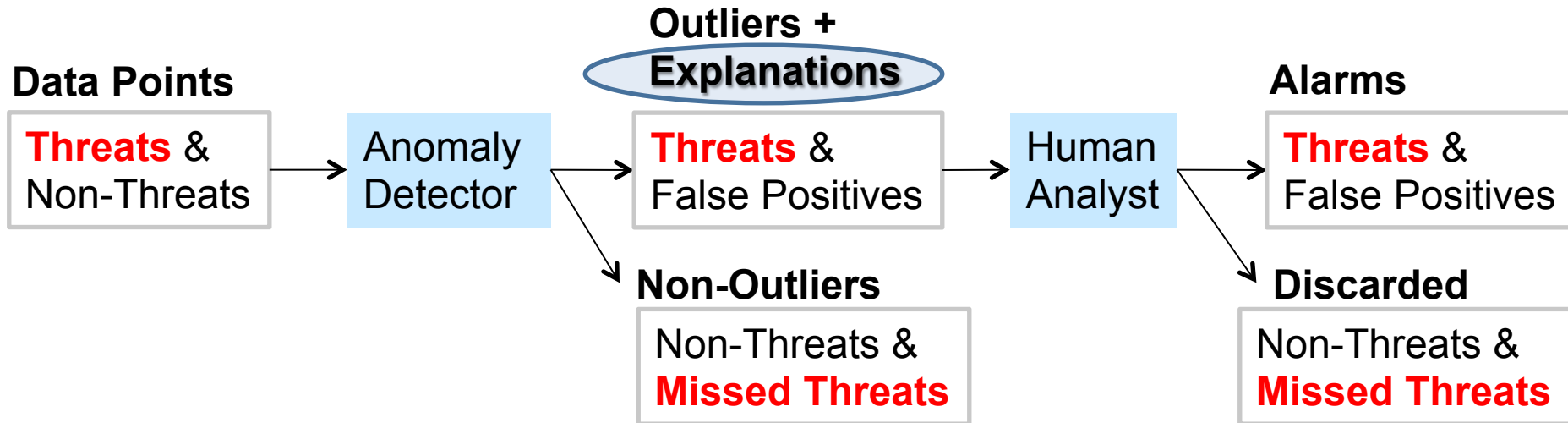
Analyst can focus on information related to explanation.

Anomaly Detection Pipeline



- **Sequential Feature Explanation (SFE)**: an ordering on features of an outlier prioritized by importance to anomaly detector
 - (F2, F10, F37, F26)

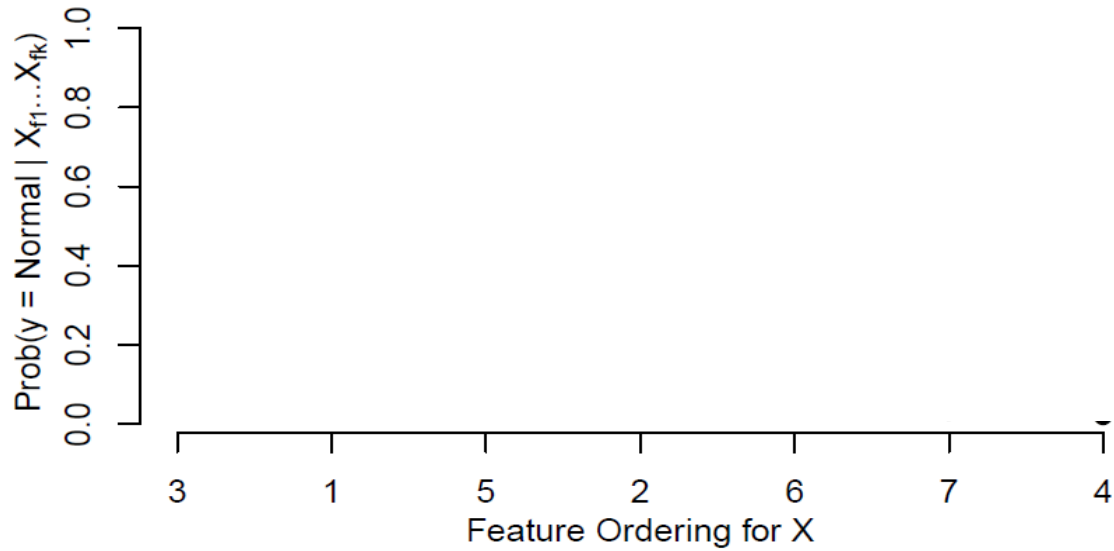
Anomaly Detection Pipeline



- **Sequential Feature Explanation (SFE)**: an ordering on features of an outlier prioritized by importance to anomaly detector
 - (F2, F10, F37, F26)
- **Protocol**: incrementally reveal features ordered by SFE until analyst makes a determination

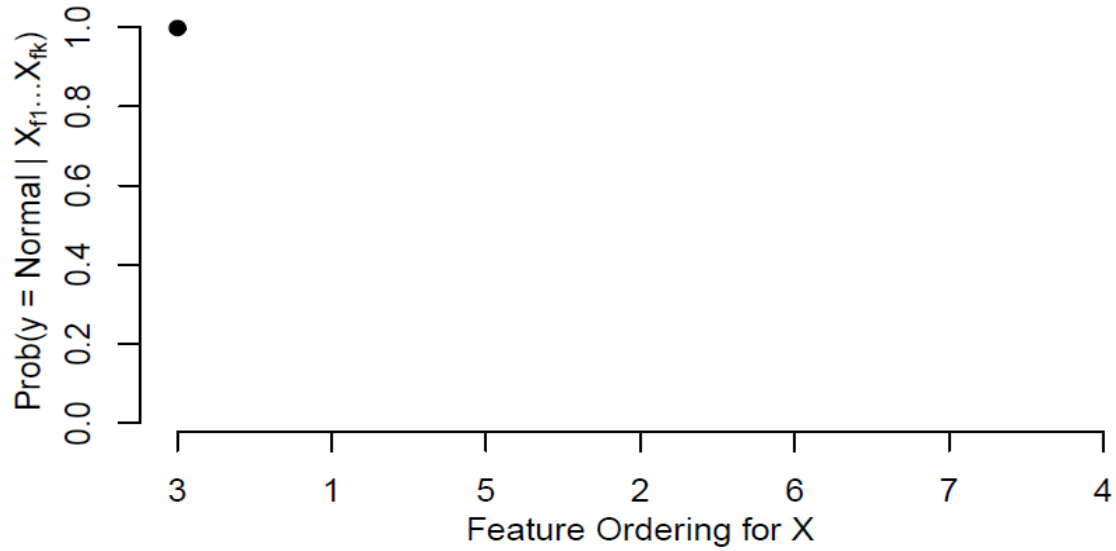
SFE Example

Analyst's belief about normality of X



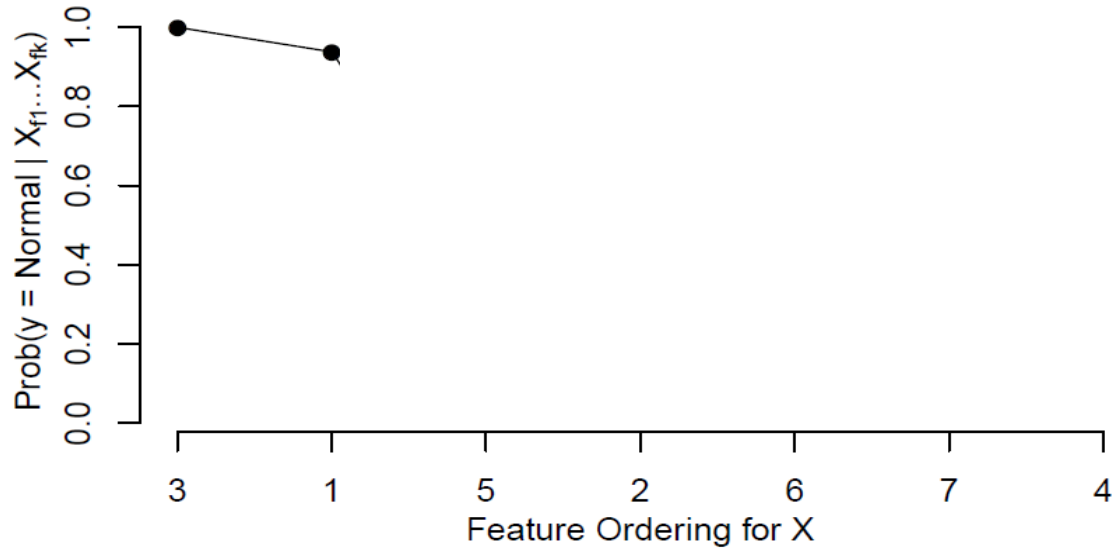
SFE Example

Analyst's belief about normality of X



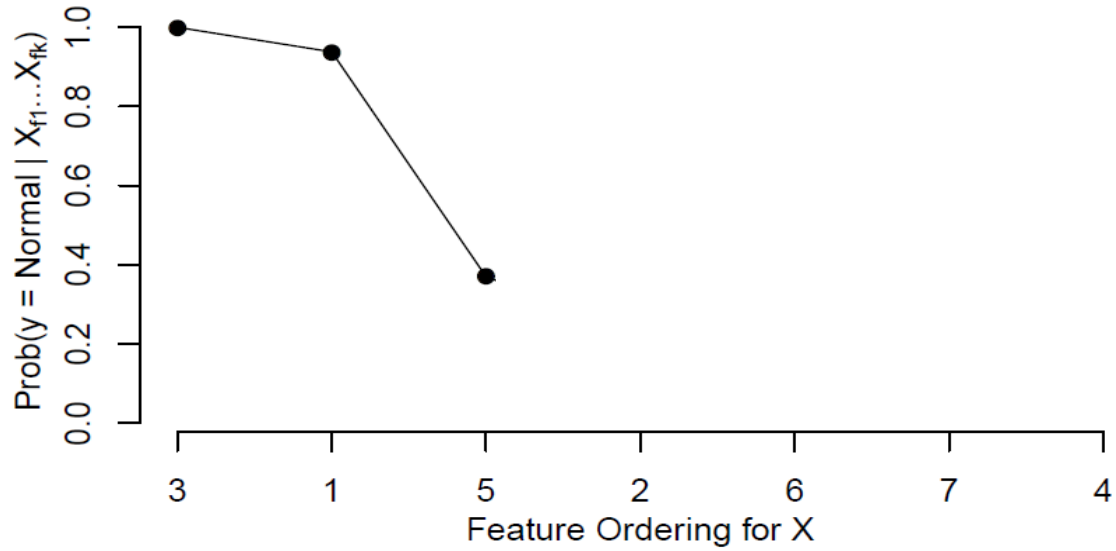
SFE Example

Analyst's belief about normality of X



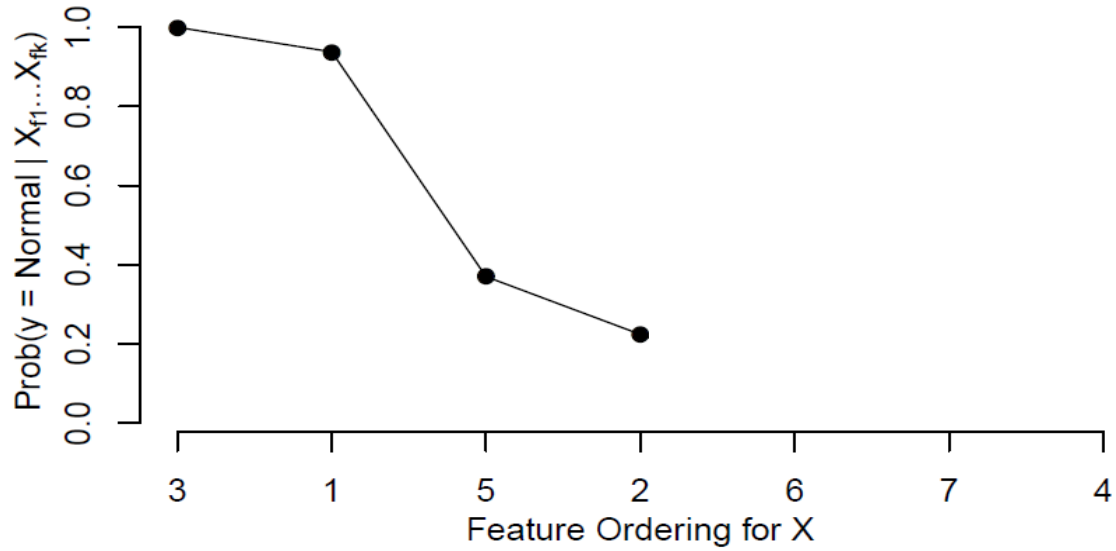
SFE Example

Analyst's belief about normality of X



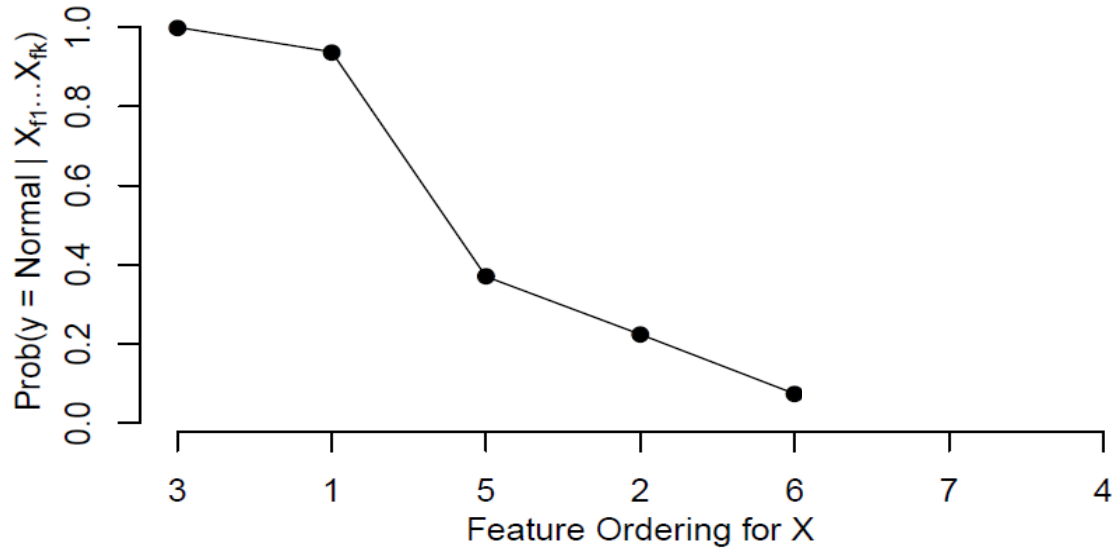
SFE Example

Analyst's belief about normality of X



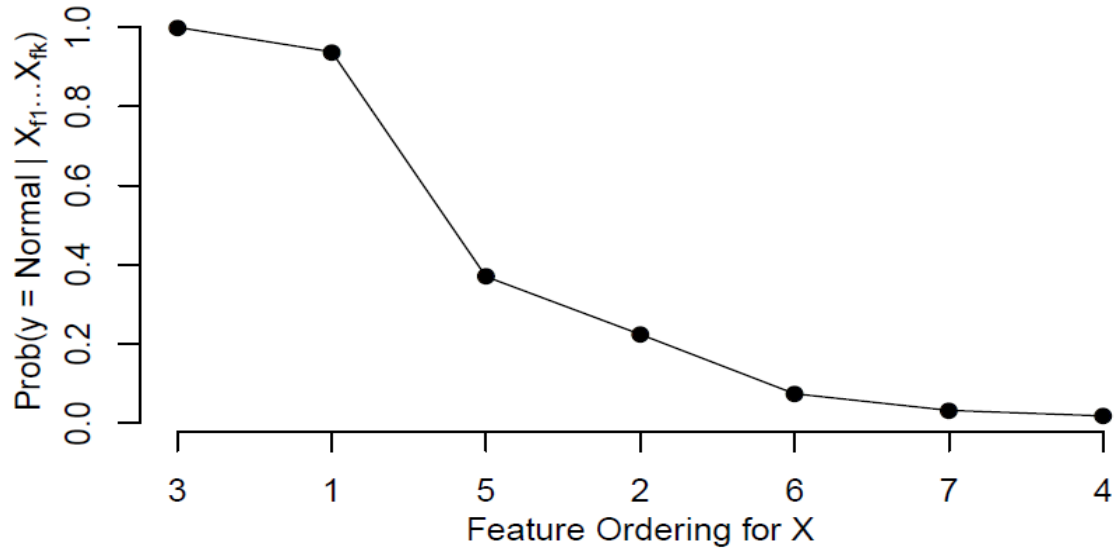
SFE Example

Analyst's belief about normality of X



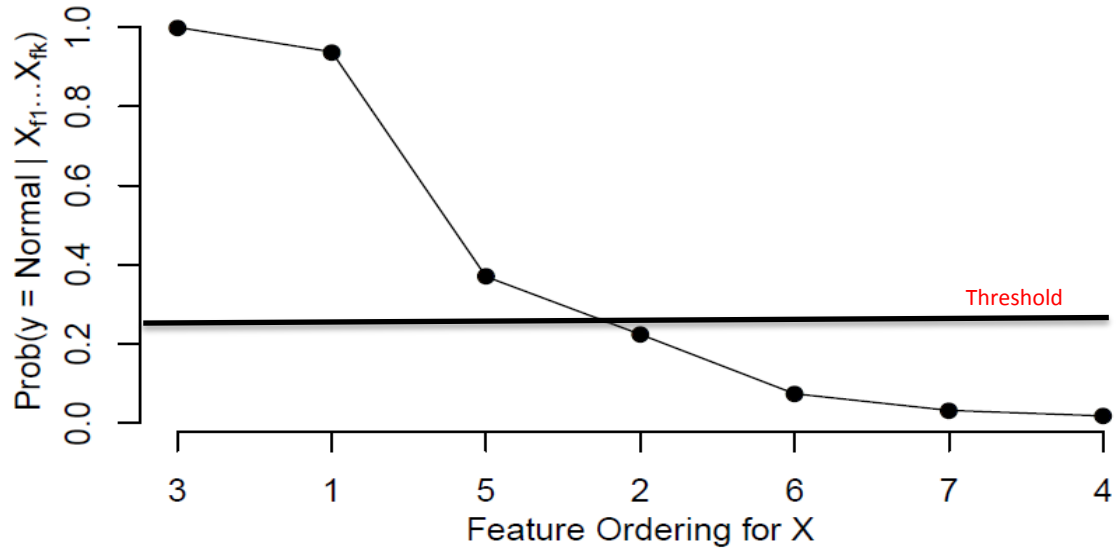
SFE Example

Analyst's belief about normality of X



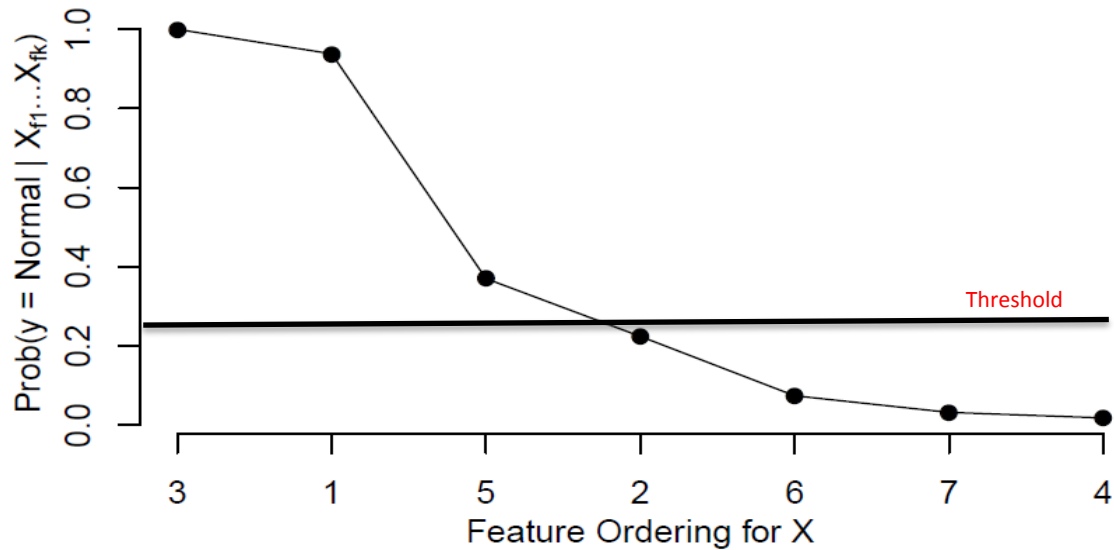
SFE Example

Analyst's belief about normality of X



SFE Example

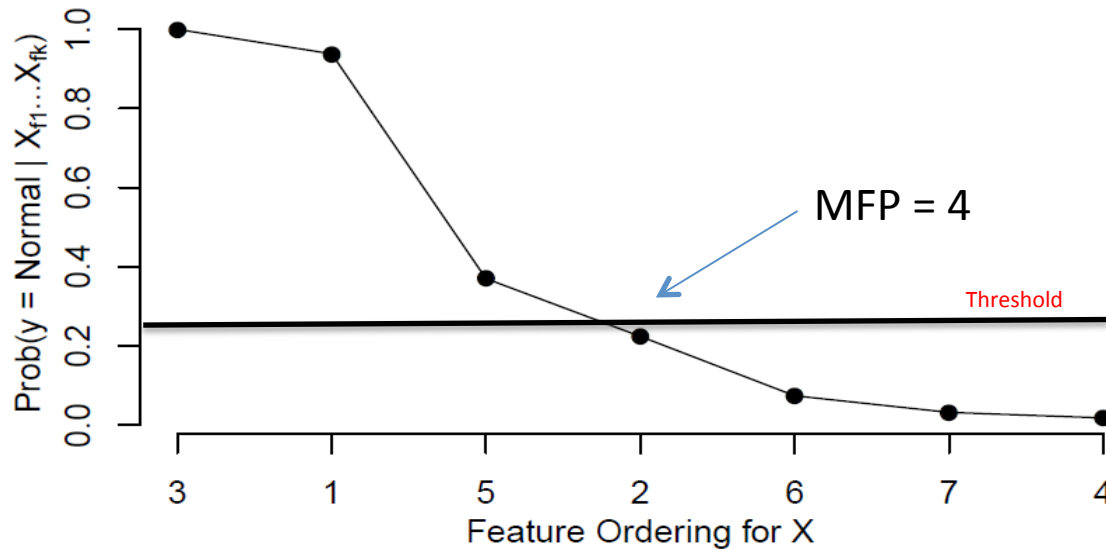
Analyst's belief about normality of X



How do we evaluate SFE quality?

SFE Example

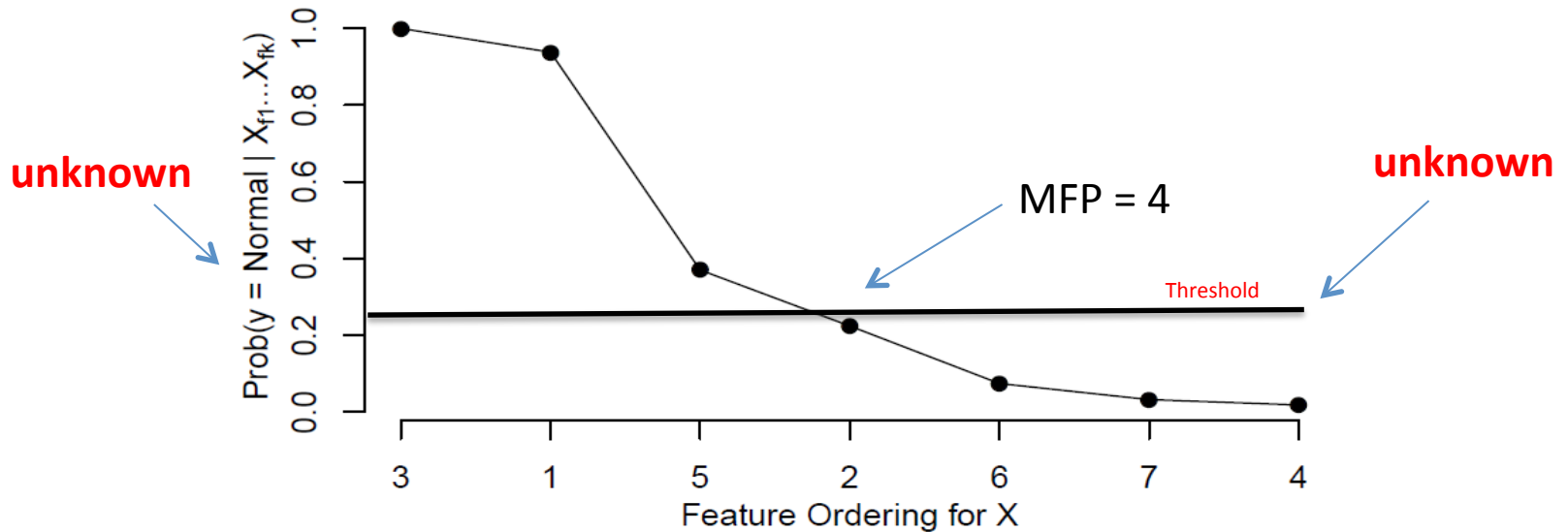
Analyst's belief about normality of X



Minimum Feature Prefix (MFP). Minimum number of features that must be revealed for the analyst to become confident that a threat is truly a threat.

Optimizing MFP

Analyst's belief about normality of X

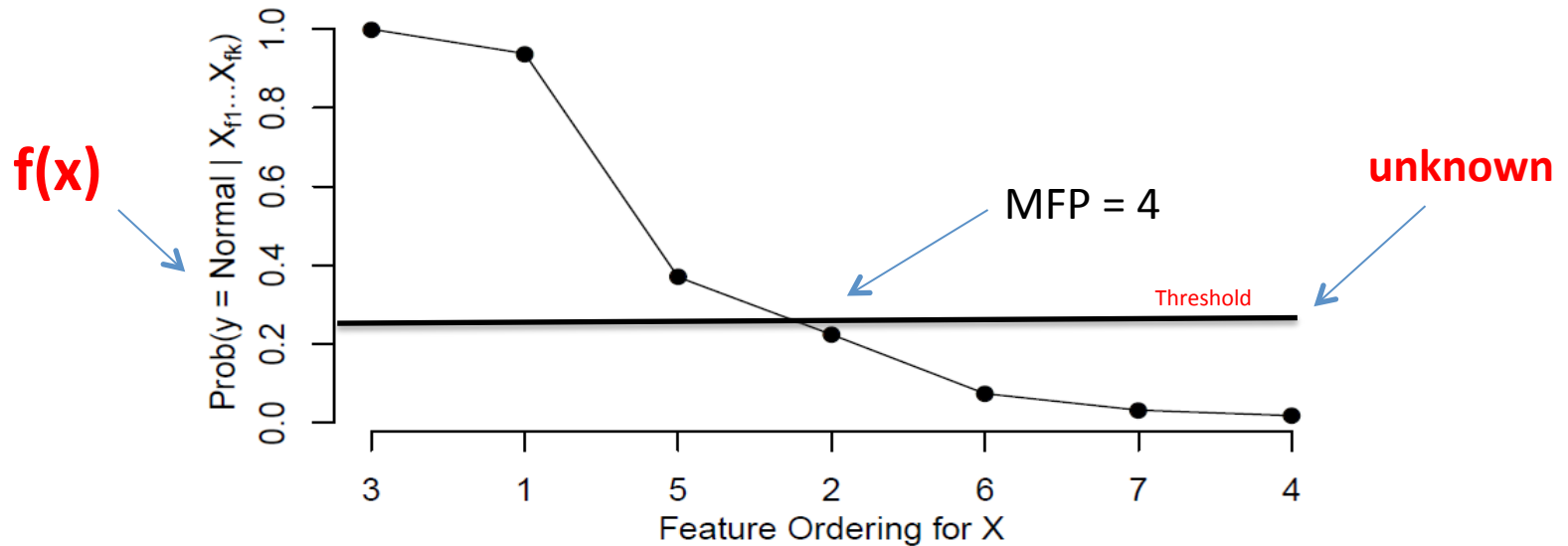


~~Ideal Objective: compute SFE with minimum MFP~~

But We don't know the analyst belief model or threshold !

Optimizing MFP

Analyst's belief about normality of X

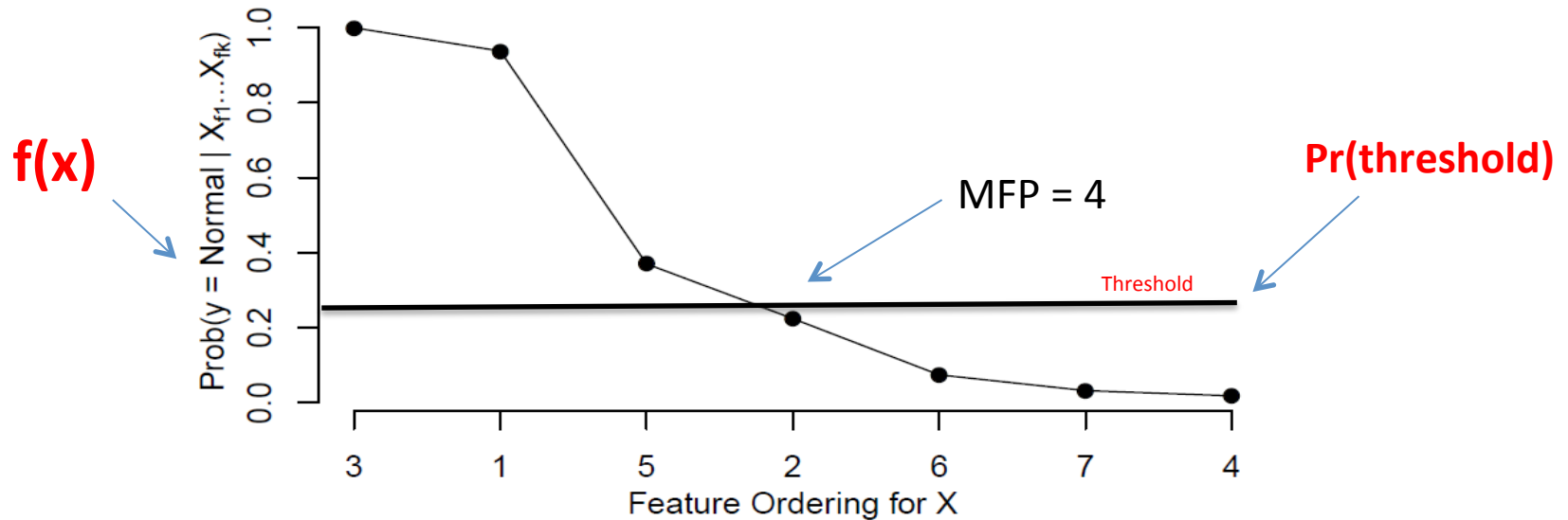


~~Ideal Objective: compute SFE with minimum MFP~~

Assumption 1: analyst's beliefs modeled by learned density $f(x)$

Optimizing MFP

Analyst's belief about normality of X



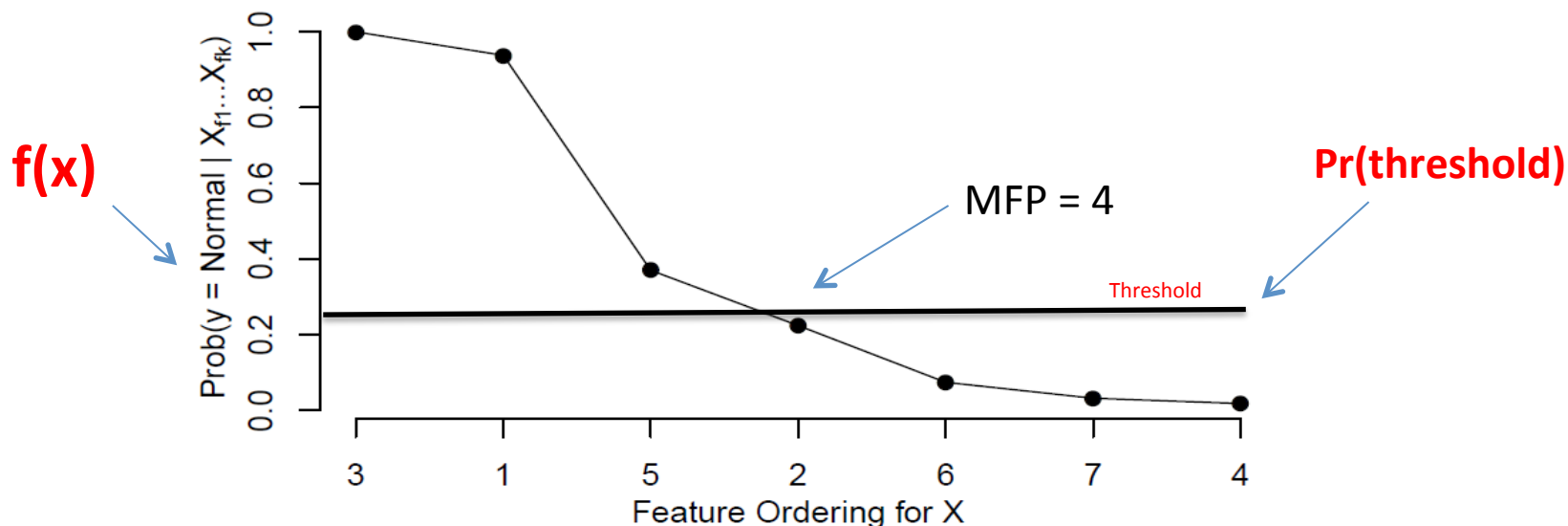
~~Ideal Objective: compute SFE with minimum MFP~~

Assumption 1: analyst's beliefs modeled by learned density $f(x)$

Assumption 2: distribution $\text{Pr}(\text{threshold})$ over possible thresholds

Optimizing MFP

Analyst's belief about normality of X



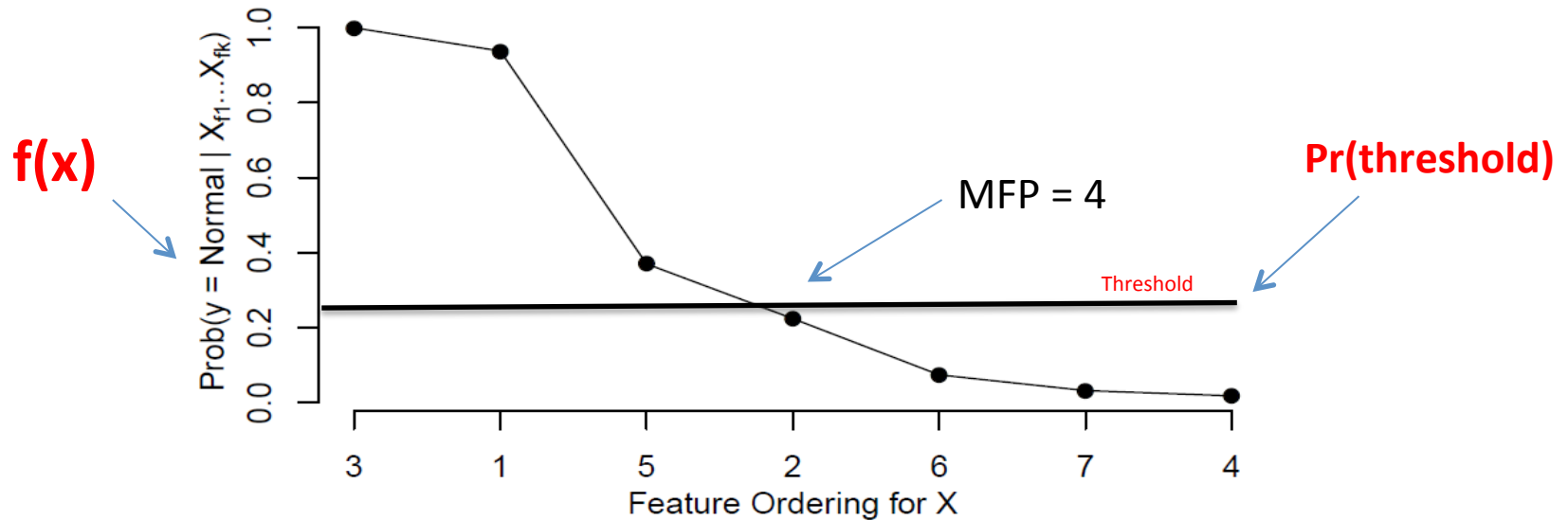
~~Ideal Objective: compute SFE with minimum MFP~~

Assumption 1: analyst's beliefs modeled by learned density $f(x)$

Assumption 2: distribution $\text{Pr}(\text{threshold})$ over possible thresholds

Optimizing MFP

Analyst's belief about normality of X



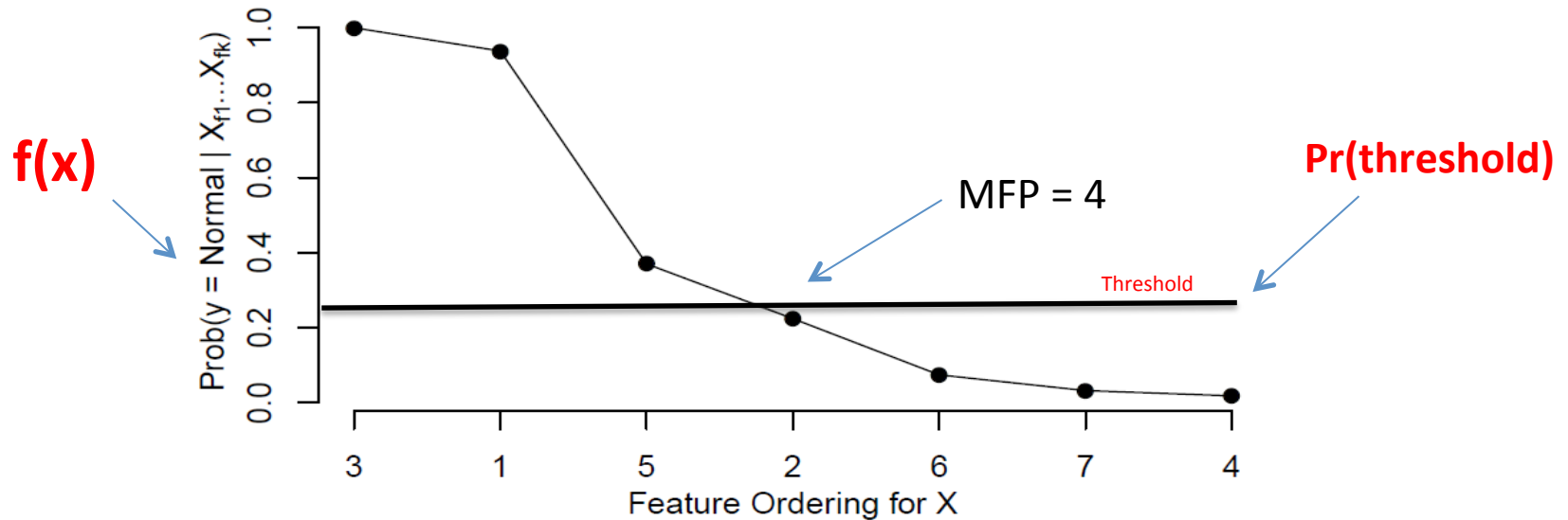
Realizable Objective: compute SFE with minimum expected MFP under assumptions 1 and 2

Assumption 1: analyst's beliefs modeled by learned density $f(x)$

Assumption 2: distribution $\text{Pr}(\text{threshold})$ over possible thresholds

Optimizing MFP

Analyst's belief about normality of X

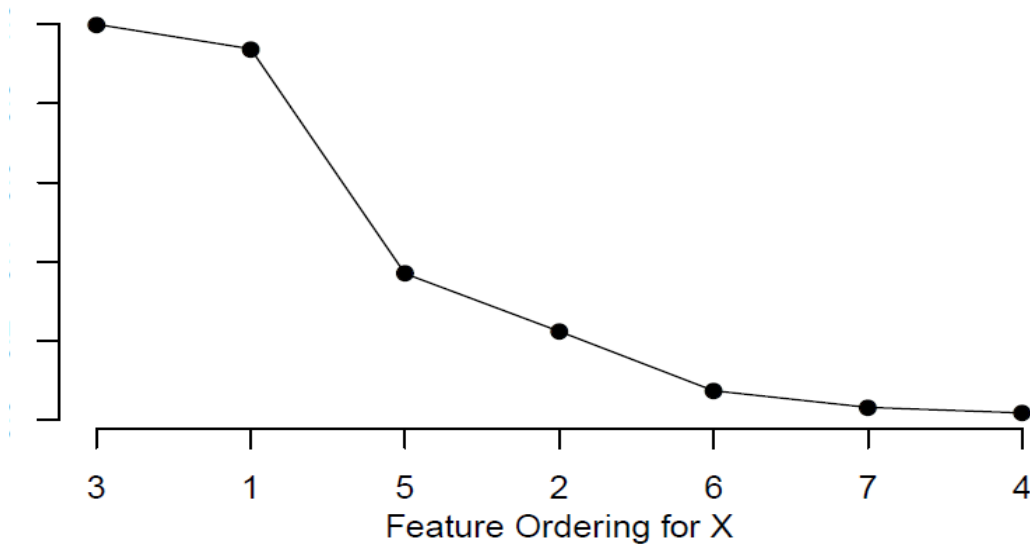


Realizable Objective: compute SFE with minimum expected MFP under assumptions 1 and 2

NP-hard problem

Not Covered Today: branch and bound optimization procedure

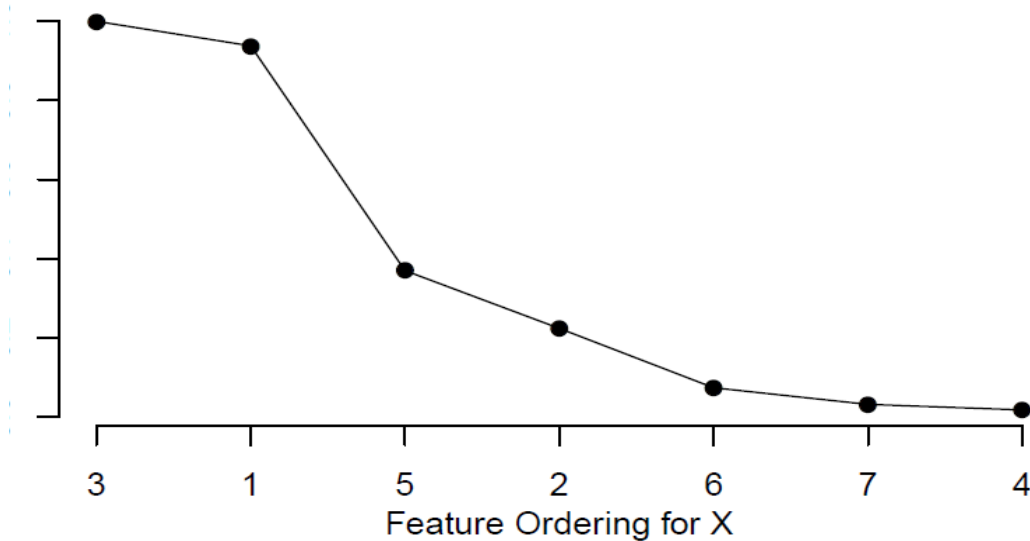
Greedy Optimization: Sequential Marginal



Sequential Marginal:

- Choose First feature i that minimizes $f(x \downarrow i)$

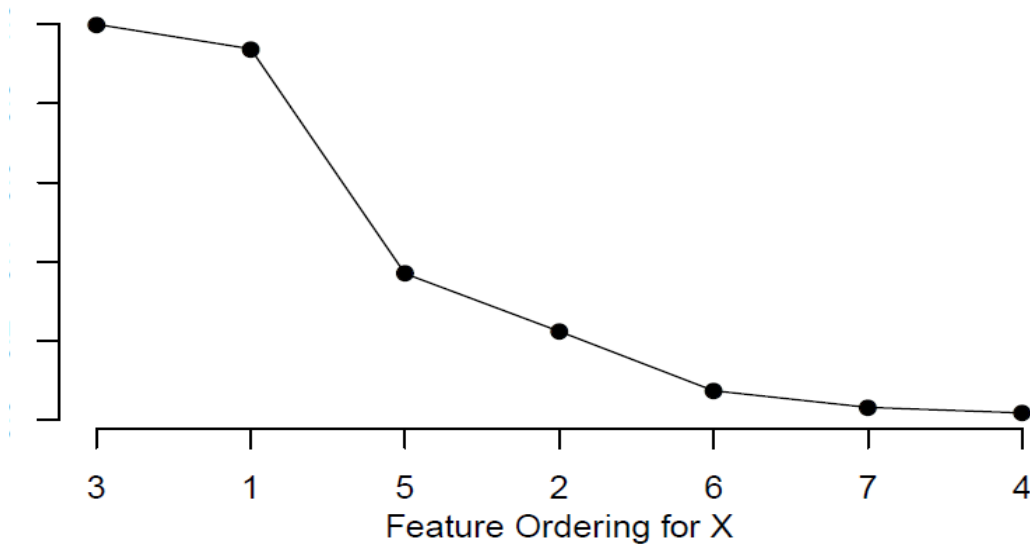
Greedy Optimization: Sequential Marginal



Sequential Marginal:

- Choose First feature i that minimizes $f(x \downarrow i)$
- Choose Second feature j that minimizes $f(x \downarrow i, x \downarrow j)$

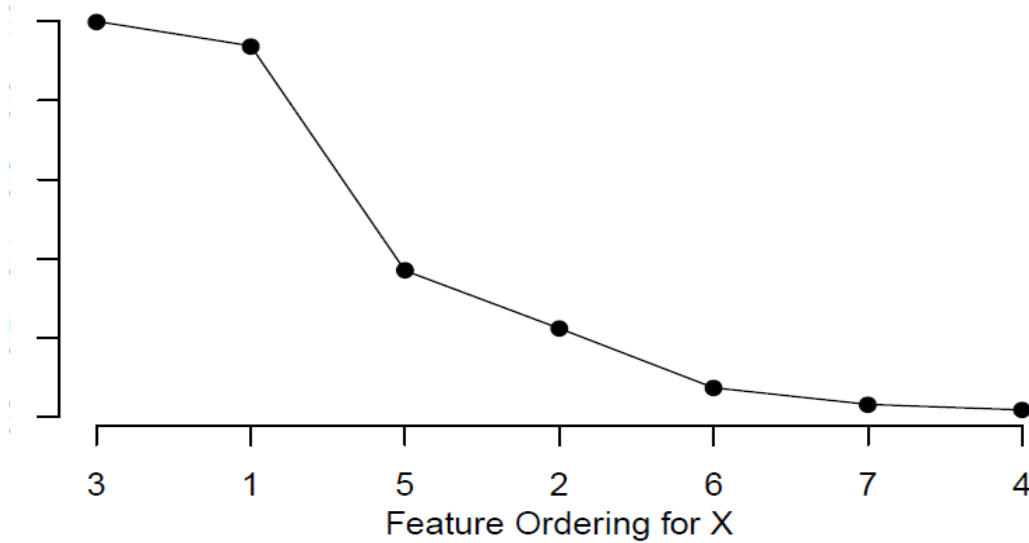
Greedy Optimization: Sequential Marginal



Sequential Marginal:

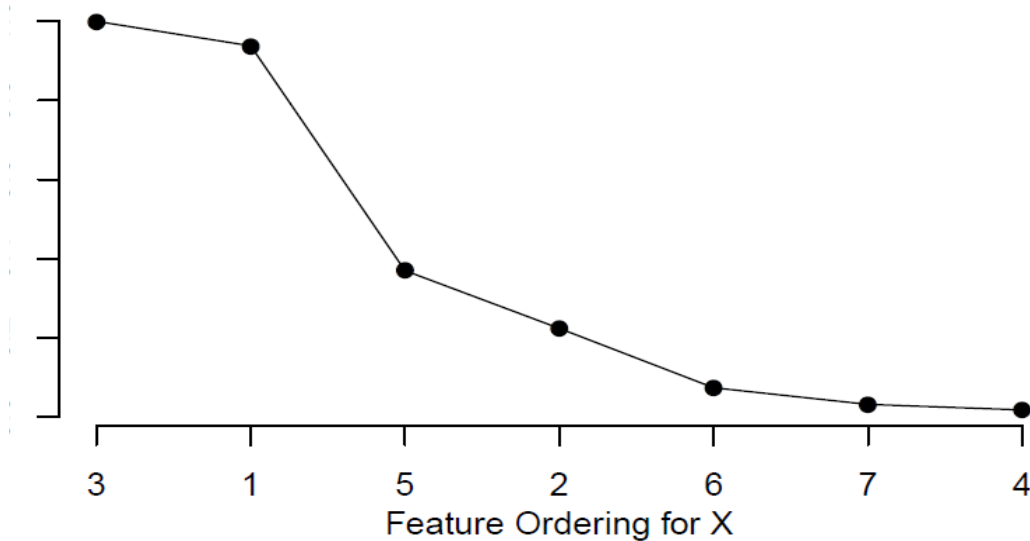
- Choose First feature i that minimizes $f(x \downarrow i)$
- Choose Second feature j that minimizes $f(x \downarrow i, x \downarrow j)$
-

Greedy Optimization: Independent Marginal



Independent Marginal: computationally cheaper

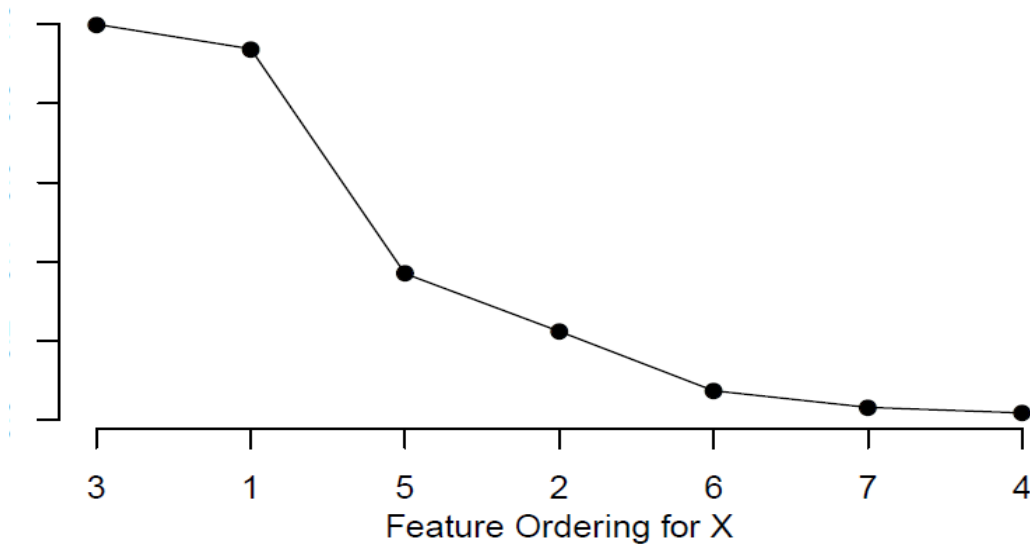
Greedy Optimization: Independent Marginal



Independent Marginal: computationally cheaper

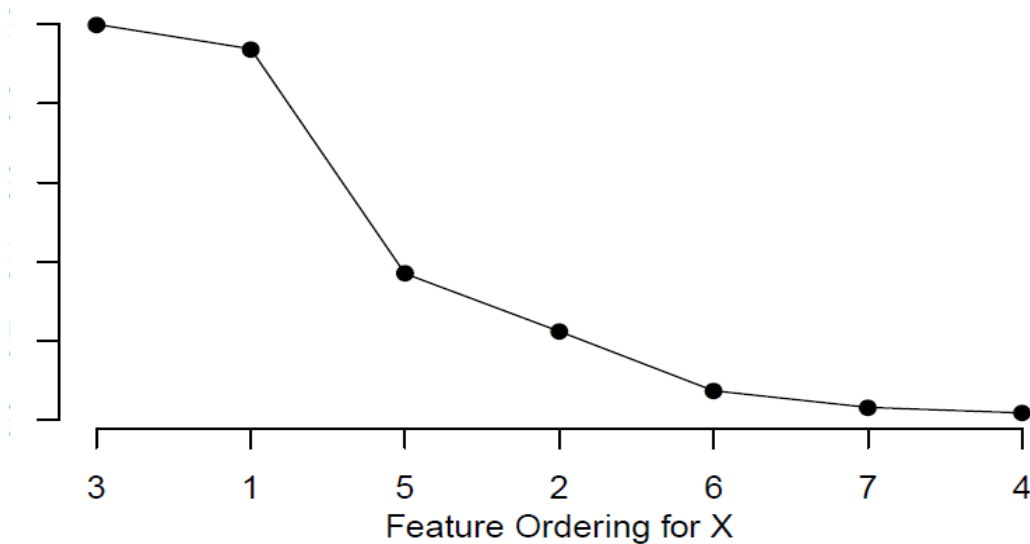
- Order features according to increasing $f(x \downarrow i)$
- I.e. order according to independent anomalousness of each feature

Greedy Optimization: Independent Dropout



Independent Dropout: inspired by [Robnik et al., 2008] for computing supervised learning explanations

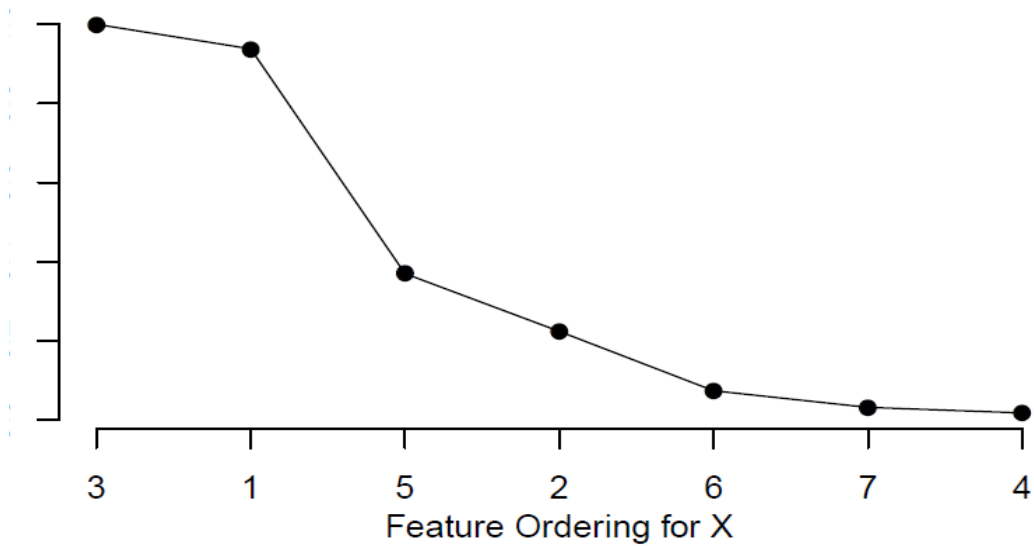
Greedy Optimization: Independent Dropout



Independent Dropout: inspired by [Robnik et al., 2008] for computing supervised learning explanations

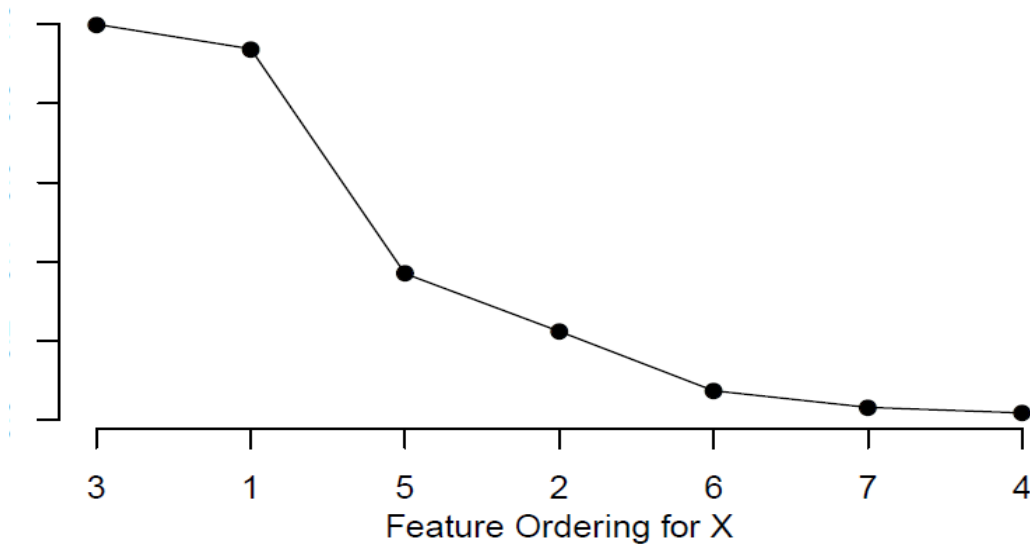
- Order features according to decreasing $f(x \downarrow -i)$
- I.e. order according to how much more normal x looks after removing the feature

Greedy Optimization: Sequential Dropout



Sequential Dropout:

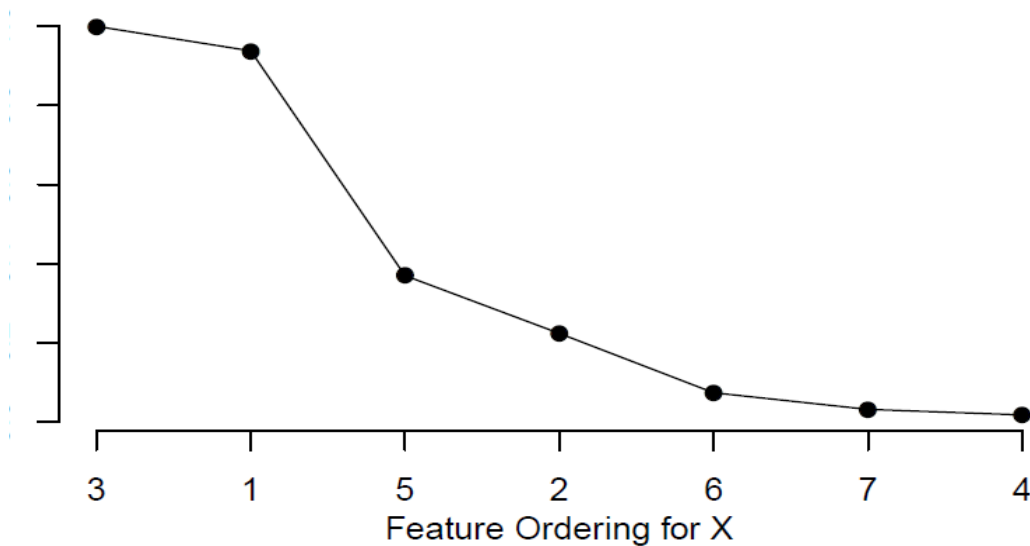
Greedy Optimization: Sequential Dropout



Sequential Dropout:

- Select first feature i as one that maximizes $f(x \downarrow -i)$

Greedy Optimization: Sequential Dropout

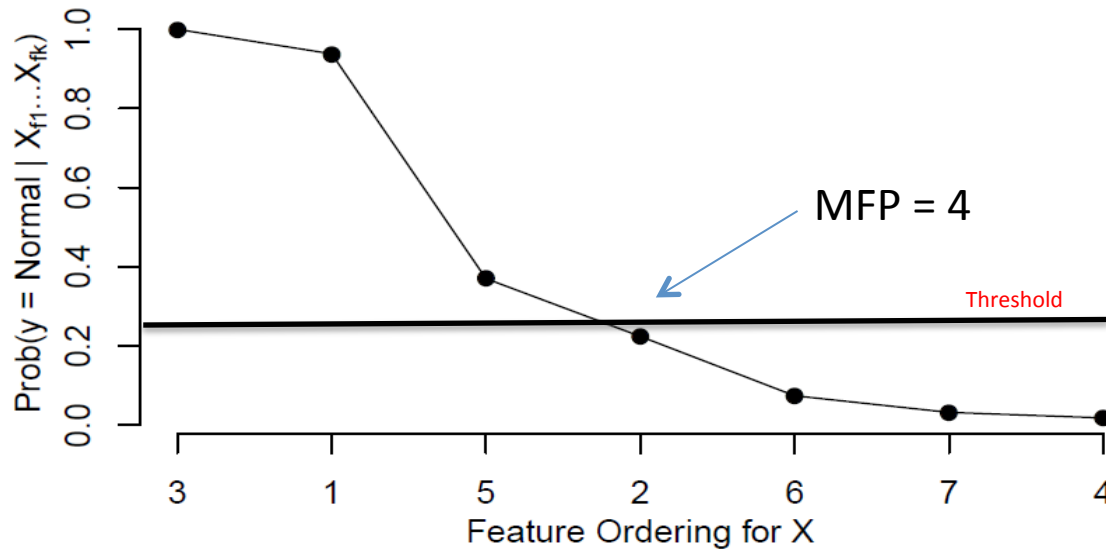


Sequential Dropout:

- Select first feature i as one that maximizes $f(x \downarrow -i)$
- Select second feature j as one that maximizes $f(x \downarrow -i-j)$
-

Evaluating SFEs

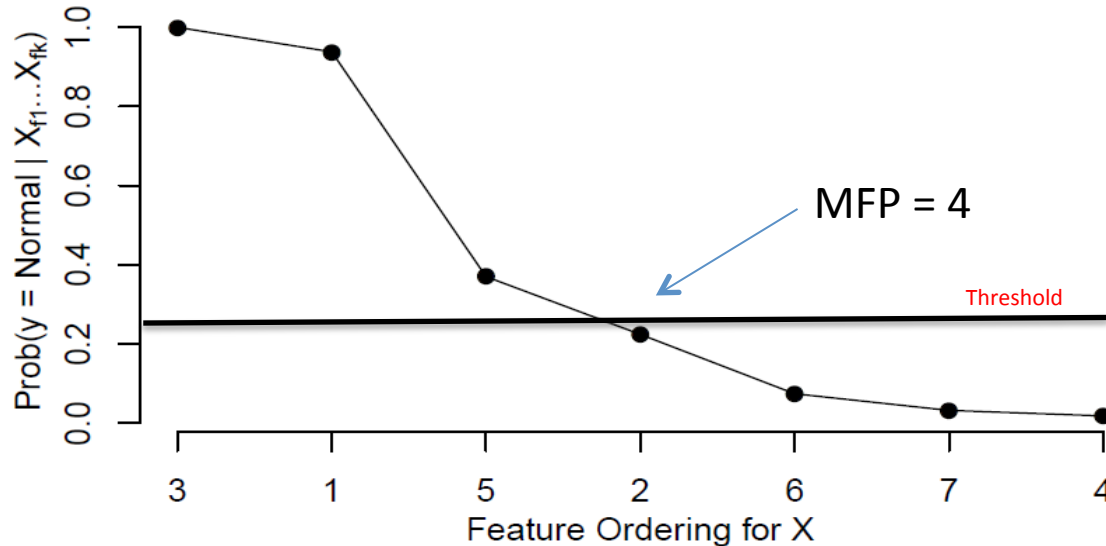
Analyst's belief about normality of X



Problem: Evaluating an SFE requires access to an analyst, but we can't run large scale experiments with real analysts

Evaluating SFEs

Analyst's belief about normality of X

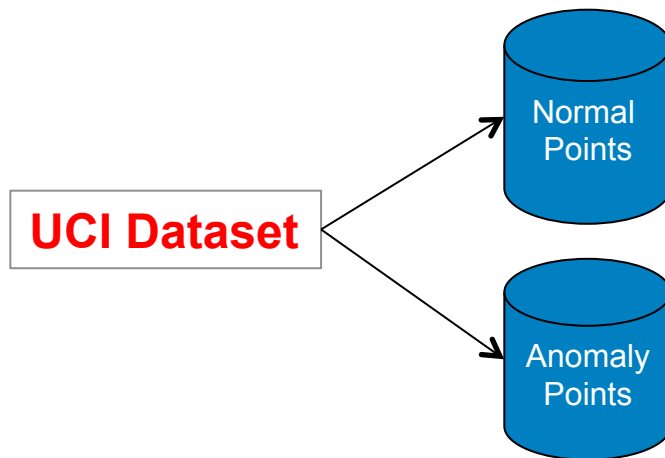


Problem: Evaluating an SFE requires access to an analyst, but we can't run large scale experiments with real analysts

Solution: Construct simulated analyst for anomaly detection benchmarks

Evaluating Explanations

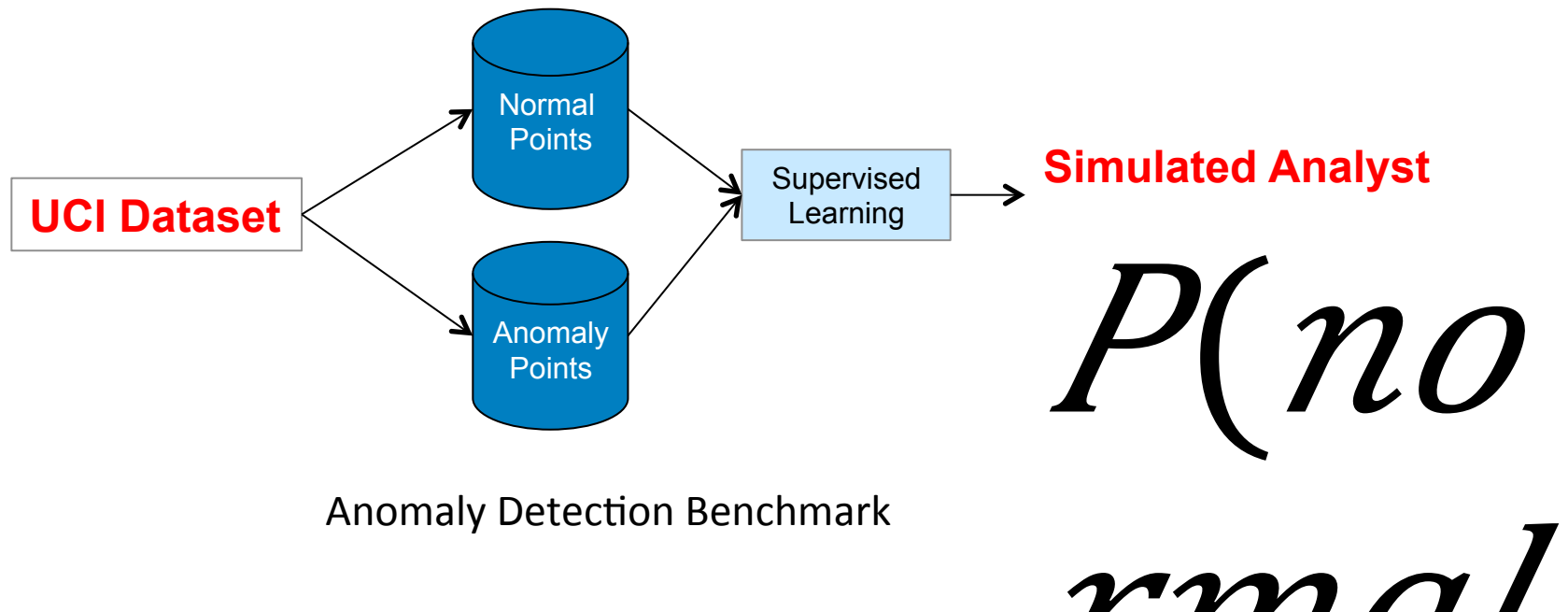
- Start with anomaly detection benchmarks constructed from UCI supervised learning data set [Emmott et al., 2013]
 - Each benchmark has known anomaly and normal classes



Anomaly Detection Benchmark

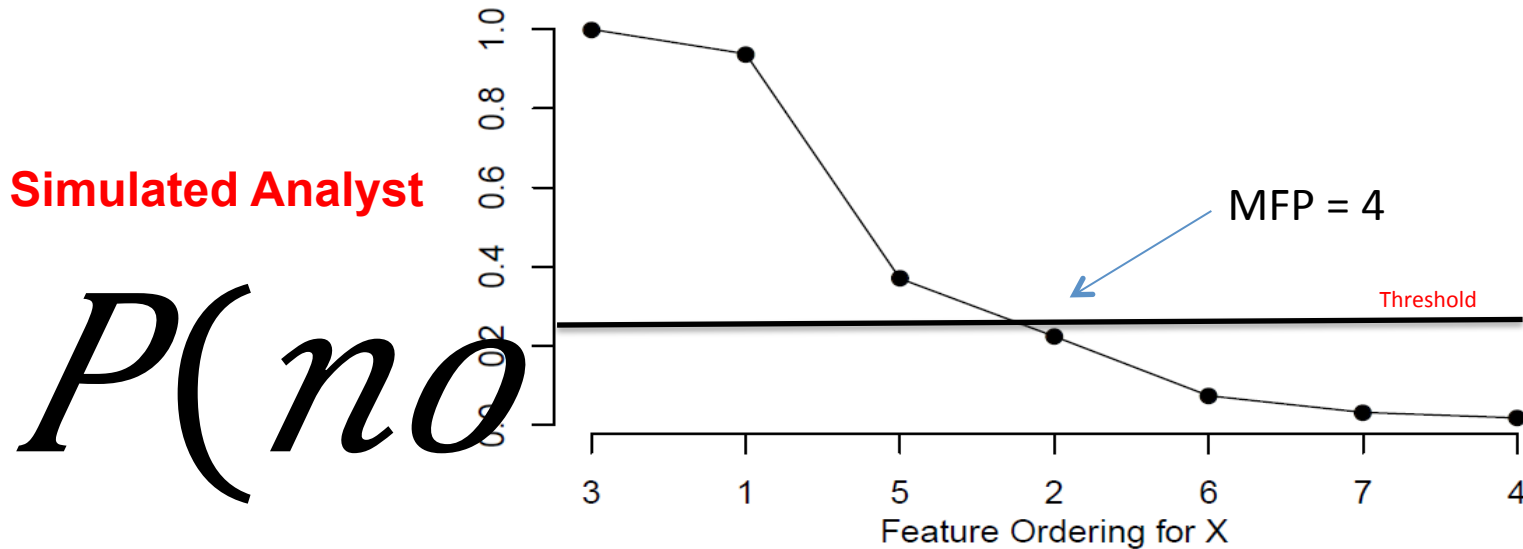
Evaluating Explanations

- Start with anomaly detection benchmarks constructed from UCI supervised learning data set [Emmott et al., 2013]
 - Each benchmark has known anomaly and normal classes
- Learn a classifier $P(\text{normal} \mid \mathbf{x})$ to predict normal vs. anomalous for any feature subset
 - Can serve as a simulated analyst



Evaluating SFEs

Analyst's belief about normality of X



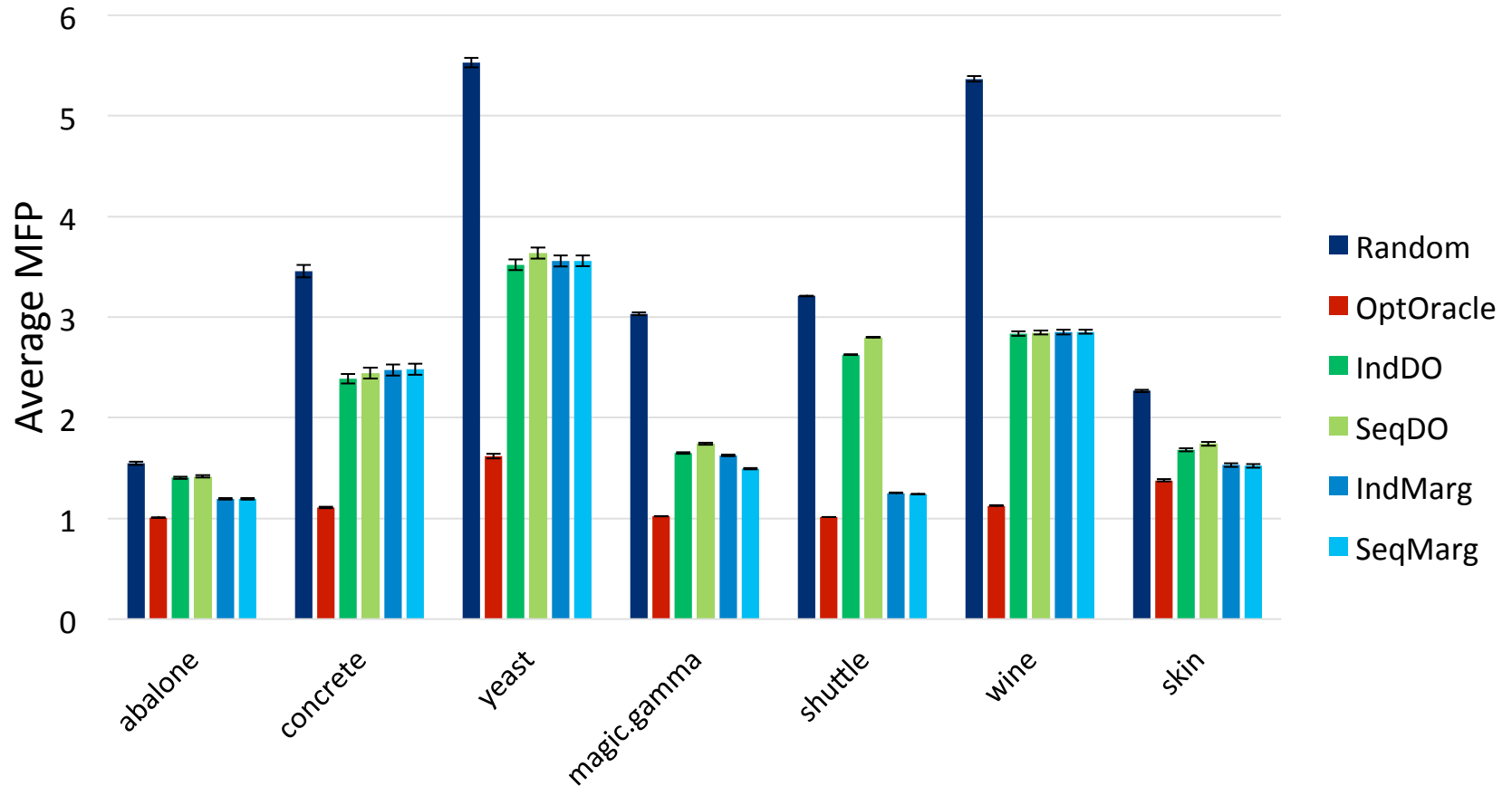
$P(\text{no}$

rma

Evaluation Metric : expected MFP of simulated analyst

Use reasonable distribution over thresholds.

Results of Explanations for EGMM



Use an ensemble of GMMs (EGMM) as the learned density $f(x)$

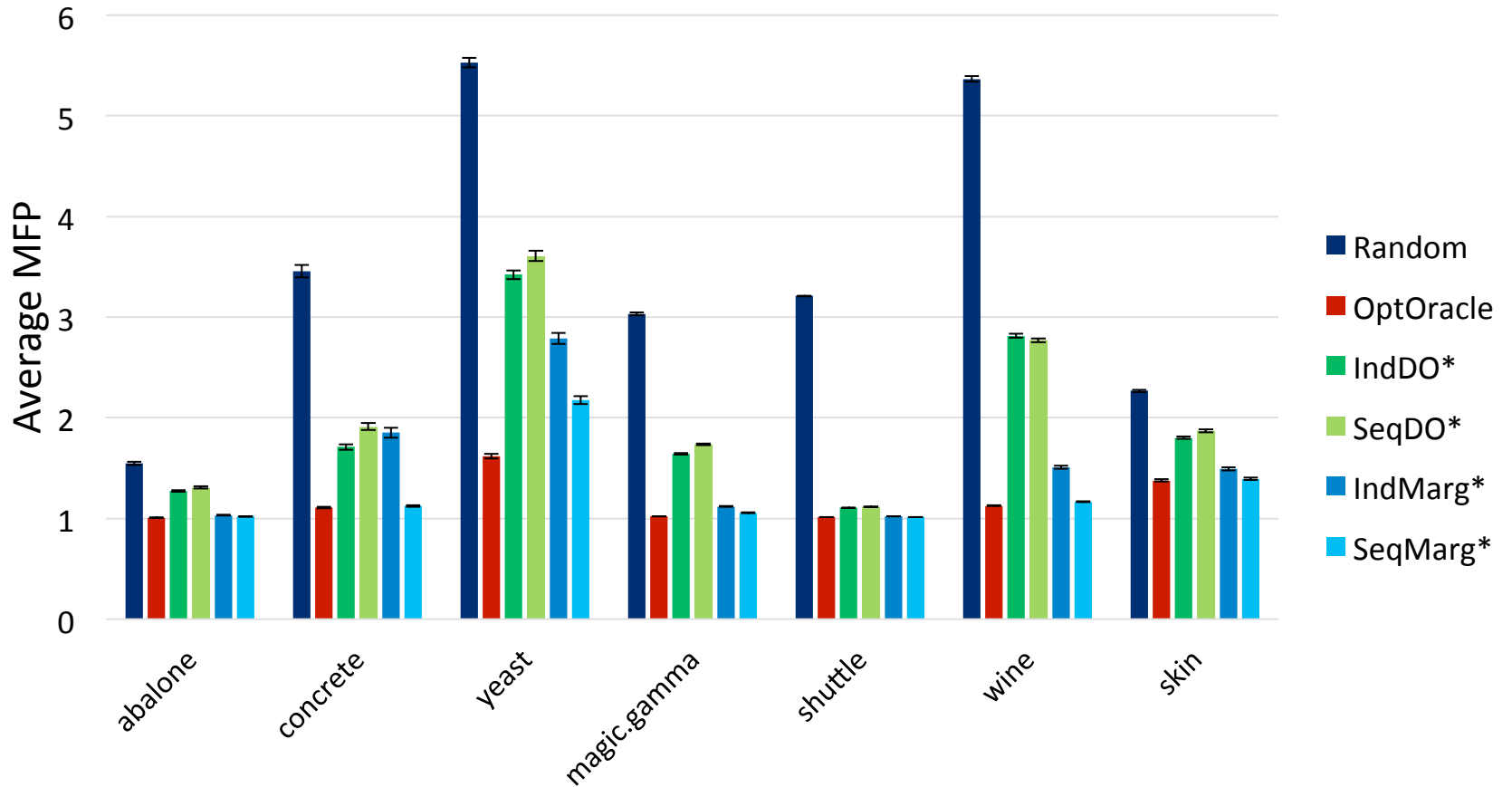
Oracle Experiments

Explanation evaluation depends on two factors:

1. Quality of $f(x)$
 - How well does $f(x)$ match true analyst?
2. Quality of explanation computation

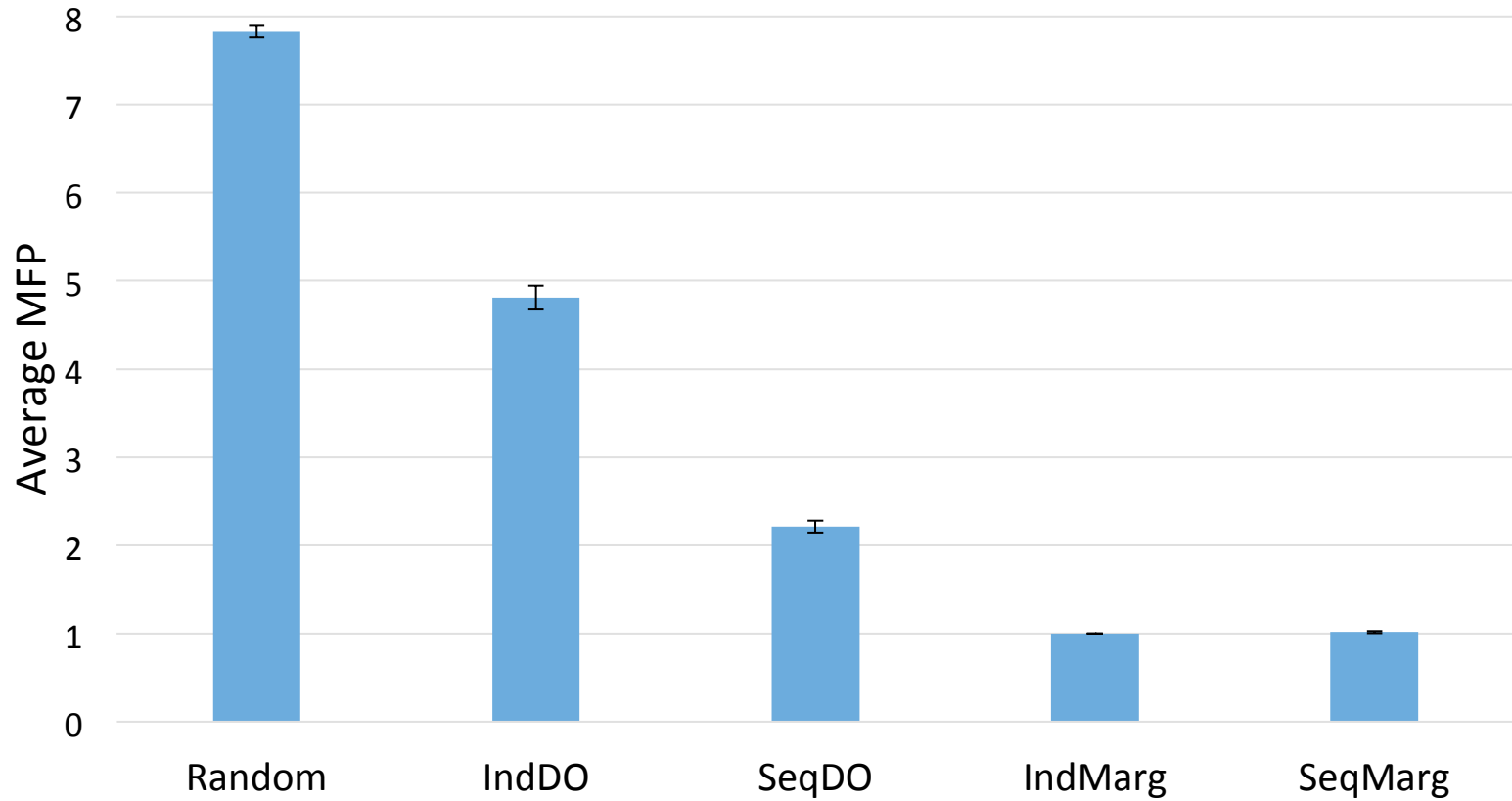
To assess (2) we run experiments that replace $f(x)$ with ground truth analyst

Results of Explanations for Oracle Detector



Result on KDDCup99 Dataset

Result on KDDCup99 Dataset



Key Observations from the Experiments

- All methods significantly beat random

Key Observations from the Experiments

- All methods significantly beat random
- Marginal methods no worse and sometimes better than dropout

Key Observations from the Experiments

- All methods significantly beat random
- Marginal methods no worse and sometimes better than dropout
- Independent marginal is nearly as good as sequential marginal

Key Observations from the Experiments

- All methods significantly beat random
- Marginal methods no worse and sometimes better than dropout
- Independent marginal is nearly as good as sequential marginal
 - But sequential is significantly better in oracle experiments
- The “weaker signals” produced by the Dropout methods when taking early decisions makes it less robust compare to the Marginal methods

Summary

- Reducing effort of analyst to detect threats can reduce the analyst miss rate

Summary

- Reducing effort of analyst to detect threats can reduce the analyst miss rate
- Proposed sequential feature explanations to guide analyst investigation

Summary

- Reducing effort of analyst to detect threats can reduce the analyst miss rate
- Proposed sequential feature explanations to guide analyst investigation
- Proposed an evaluation framework for explanations

Summary

- Reducing effort of analyst to detect threats can reduce the analyst miss rate
- Proposed sequential feature explanations to guide analyst investigation
- Proposed an evaluation framework for explanations
- Designed 4 greedy explanation methods and evaluated

Summary

- Reducing effort of analyst to detect threats can reduce the analyst miss rate
- Proposed sequential feature explanations to guide analyst investigation
- Proposed an evaluation framework for explanations
- Designed 4 greedy explanation methods and evaluated
- **Preferred Method:** sequential marginal

Future Work

- Further evaluations
 - Additional anomaly detectors (e.g. with PCA applied)
 - Larger feature spaces
- Evaluate non-greedy algorithms
 - Branch-and-Bound
- Anomaly exoneration
- Alternative types of explanations

Questions

SFE Calculation

- We assume, for every feature subset s there exists a particular threshold τ such that for any instance x : $f(x \downarrow s) < \tau$ implies x is an anomaly
- To find optimal *SFE* we first define the *MFP* of a *SFE* E for an instance x :
$$MFP(x, E, \tau(E)) = \min\{ i : f(x \downarrow E \downarrow 1:i) < \tau \downarrow i(E) \}$$

Where

$f(\cdot)$ is the density function

$\tau(E)$ is the set of thresholds, where $\tau \downarrow i(E)$ is a random variable corresponding to the feature subset $E \downarrow 1:i$

SFE Calculation

- Expected *MFP*:

$$MFP(x,E) = E \downarrow \tau(E) [MFP(x,E,\tau(E))]$$

- Objective function for getting optimal *MFP* of x :

$$\arg \min_{\tau} E MFP(x,E)$$

- The objective function is hard to optimize, hence, we introduce two greedy methods: Marginal and Dropout, those approximately try to minimize the objective function for computing SFE

Explanation Algorithms

$f(x)$ is the learned “normal”

Sequential Marginal:

- Choose First feature i that minimizes $f(x \downarrow i)$
- Choose Second feature j that minimizes $f(x \downarrow i, x \downarrow j)$
-

Independent Marginal: computationally cheaper

- Order features according to increasing $f(x \downarrow i)$
- I.e. order according to independent anomalousness of each feature

Independent Dropout: inspired by [Robnik et al., 2008] from supervised learning

- Order features according to decreasing $f(x \downarrow -i)$
- I.e. order according to how much more normal x looks after removal

Sequential Dropout:

- Select first feature i as one that maximizes $f(x \downarrow -i)$
- Select second feature j as one that maximizes $f(x \downarrow -i-j)$
-