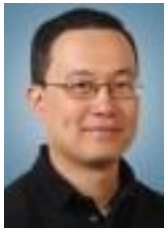
SYSTEMATIC CONSTRUCTION OF ANOMALY DETECTION BENCHMARKS FROM REAL DATA

Outlier Detection
And Description
Workshop 2013

Authors

Andrew Emmott
emmott@eecs.oregonstate.edu

Thomas Dietterich
tgd@eecs.oregonstate.edu



Weng-Keen Wong
wong@eecs.oregonstate.edu

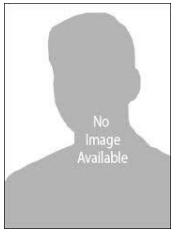
Shubhomoy Das
dassh@eecs.oregonstate.edu

Alan Fern
afern@eecs.oregonstate.edu

Staff

Jed Irvine

Mike Sander



Michael Slater

Acknowledgements

DARPA

SAIC®

Anomaly Detection At Multiple Scales

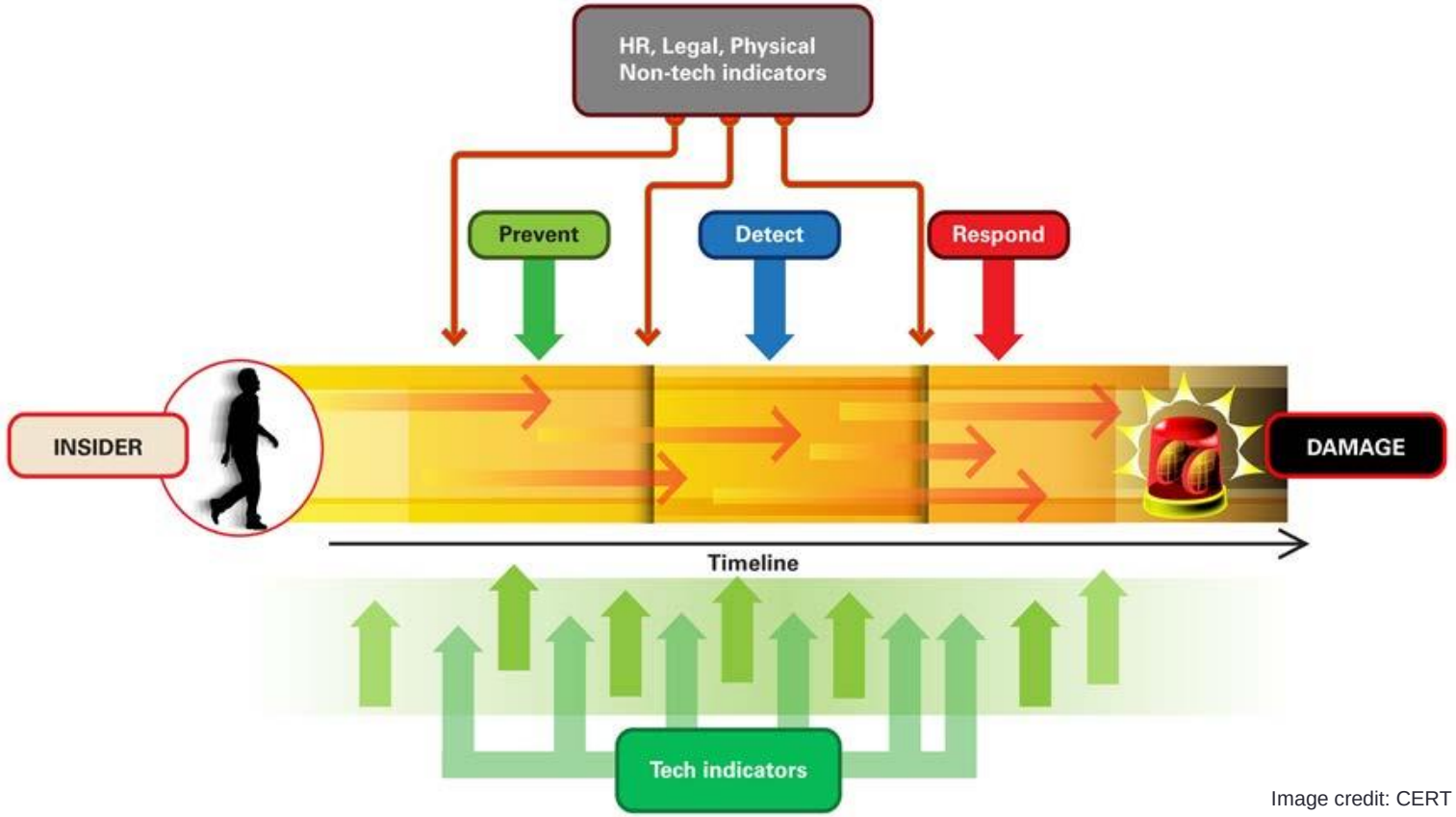


Image credit: CERT

- Application domain: Insider threat detection
- Good results but on private data; hard to publish

How are anomaly detection methods evaluated?

Synthetic Data

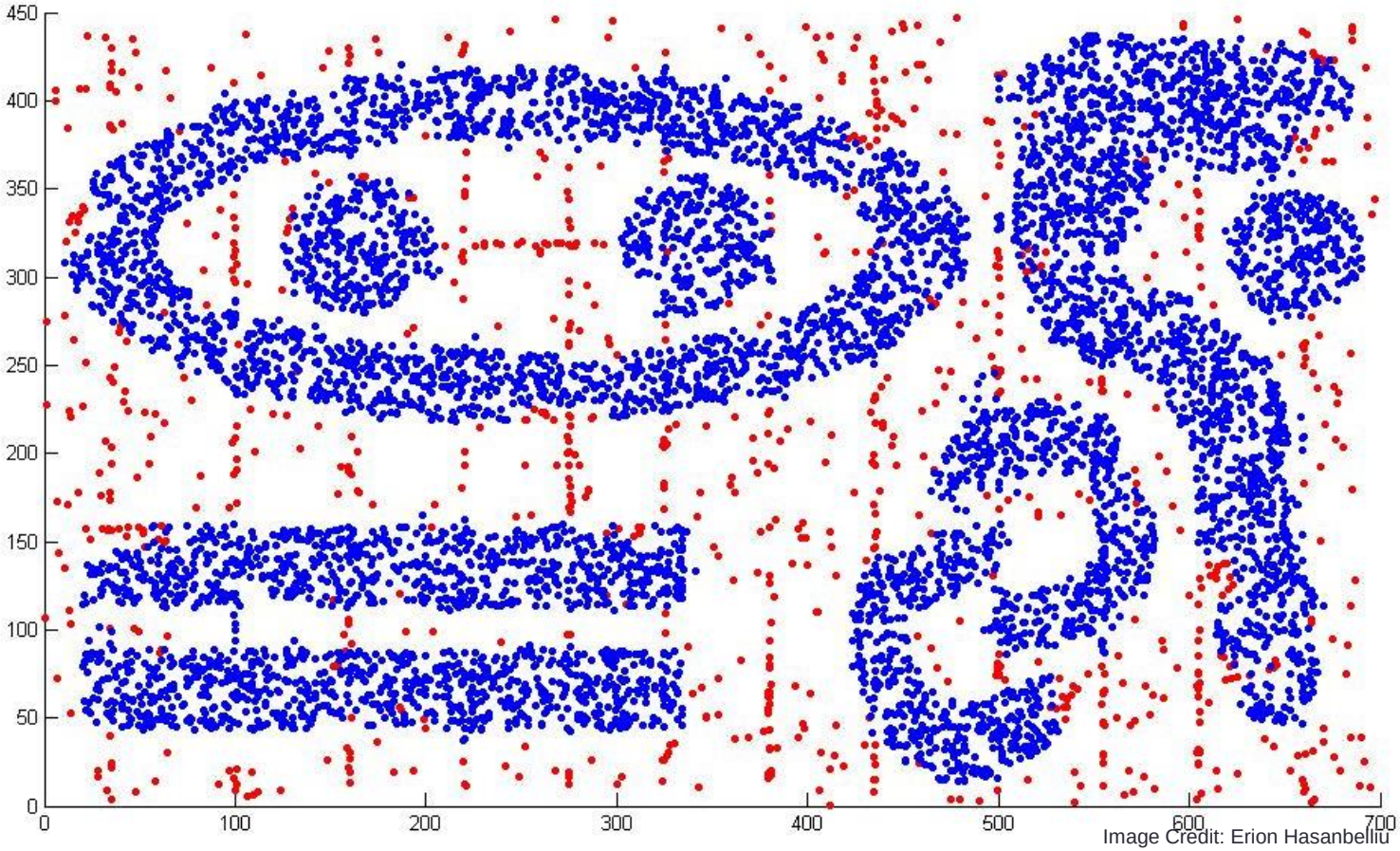


Image Credit: Erion Hasanbelliu

- Pro: Ground truth density is perfectly labeled.
- Con: Synthetic data is synthetic.

Real Data

Wine Quality:
Donor suggests extreme values as anomalies.

Abalone:
Distinguish candidate anomalies by regression value.

KDD Cup 99:
Posed as a supervised learning problem.
Intrusions are more numerous than reality.

All approaches require modifying existing data sets.

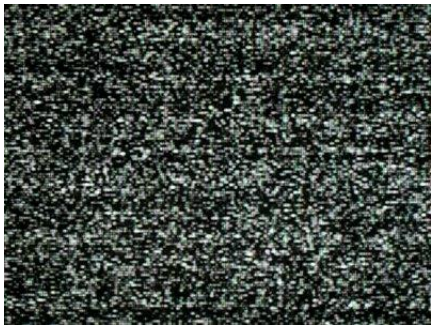
Requirements for anomaly detection benchmarks

- Points are drawn from real data.
- Anomalies are semantically distinct from normal points.
- Many benchmark datasets are needed.
- Benchmark datasets should be characterized in terms of well defined and meaningful problem dimensions that can be systematically varied.

Proposed problem dimensions

- Point difficulty – how difficult is it to distinguish the anomalies from the normal points?
    - Insider threats want to blend in.
    - Jet engine failures do not.

- Relative frequency – How much of the data is anomalous?
    - Signal failure might be relatively frequent.
    - Insider threats might be very very rare.



Proposed problem dimensions

- Semantic variation – How similar are the anomalies to each other?
    - Are they a de facto class?
    - Or are they simply not normal?
- Feature relevance/irrelevance – How well do statistical outliers in the data map to the application target?
    - An 8 foot tall man is an outlier.
    - But he is not an insider threat.

How do we enforce these requirements?

Points are drawn from real data:

We chose “mother” sets from the UC Irvine Machine Learning Repository using the following criteria:

- *Task*: Classification (binary or multi-class) or Regression. No Time-Series.
- *Instances*: At least 1000. No upper limit.
- *Features*: No more than 200. No lower limit.
- *Values*: Numeric only. Categorical features are ignored if present. No missing values, except where easily ignored.

This resulted in the selection of 19 data sets.

Anomalies are semantically distinct from normal points:

We use the semantics of the mother set to distinguish candidate nominals and candidate anomalies.

- For binary classification tasks, we are already given two semantically distinct groups.

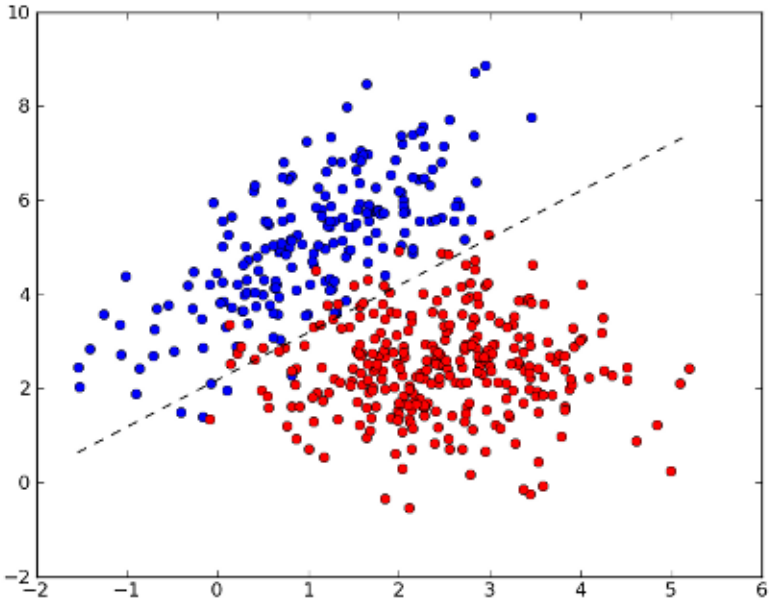


Image credit: mlpy.org

Anomalies are semantically distinct from normal points:

We use the semantics of the mother set to distinguish candidate nominals and candidate anomalies.

- For regression tasks we split the data into two groups at the median response value.

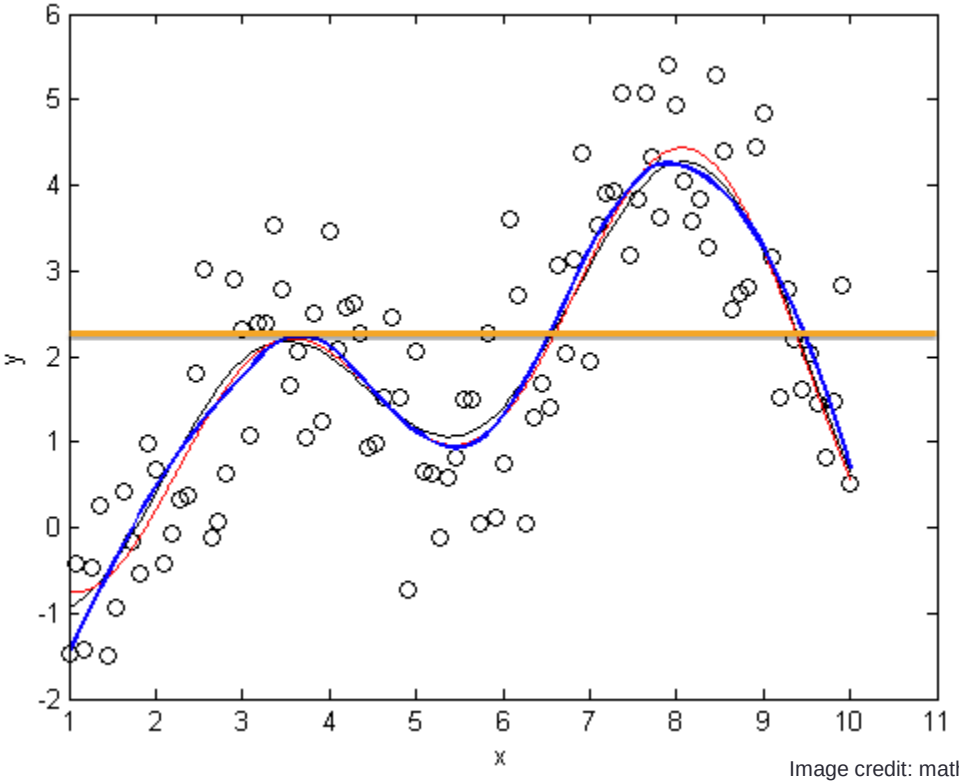


Image credit: mathworks

Anomalies are semantically distinct from normal points:

We use the semantics of the mother set to distinguish candidate nominals and candidate anomalies.

- For multi-class classification tasks we partition the classes such that we maximize the confusion between them …

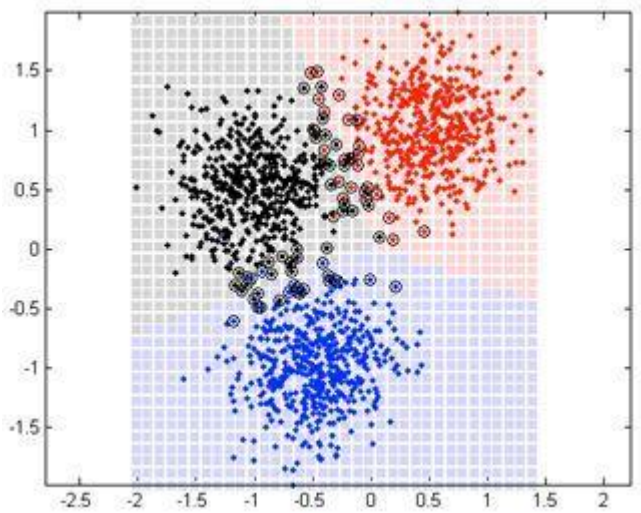


Image credit: UNITEN

Maximizing Confusion: How and Why

- Why? We need confusion between the candidate nominal and anomaly classes to ensure a variety of point difficulty, (discussed later).

- How?

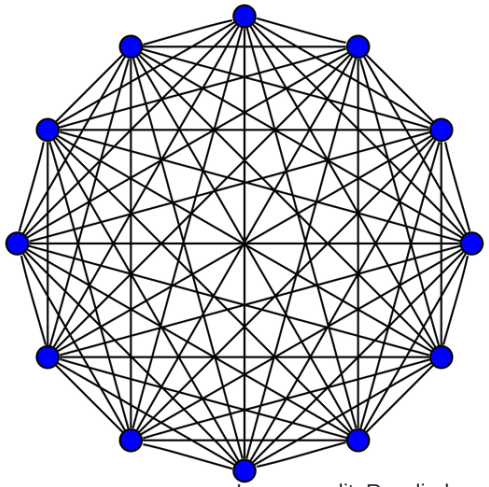


Image credit: Rosalind

  - Consider a complete graph with the classes as vertices.
  - Assign weights to the edges equal to the confusion between classes, (discussed next)
  - Find a maximum weight spanning tree on this graph.
  - 2-color the tree; the coloring is your partition.

Measuring Confusion

- We used Breimann's Random Forest algorithm to model the original multi-class problem.

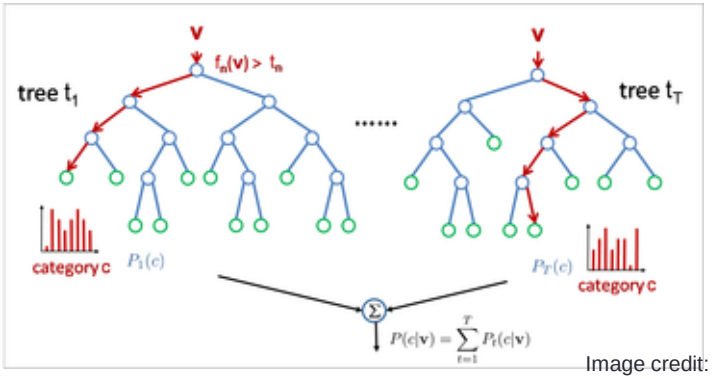


Image credit: ICCV 2009

- The forest provides a probabilistic belief about each point's membership in each class.

- All probability mass that exists between classes is summed as our confusion measure.

Many benchmark datasets are needed.

- Our methodology generated 5,120 benchmarks from the original 19 mother sets.

Benchmark datasets should be characterized in terms of well defined and meaningful problem dimensions that can be systematically varied.

- Our contribution includes a methodology for controlling and measuring **point difficulty**, **relative frequency**, and **semantic variation**.

Point Difficulty

- We model the binary version of the mother set with kernel logistic regression.
  - Note: We label the candidate anomalies as class 0, and the nominals as class 1.
- We take the response given by the KLR model on the candidate anomalies as a measure of their difficulty.

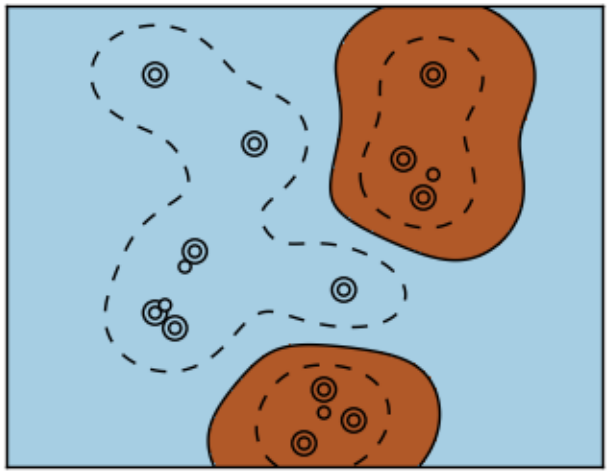


Image credit: scikit

- For this study, we bucket the candidate anomalies into four groups:

  easy $\in (0, \frac{1}{6})$ medium $\in [\frac{1}{6}, \frac{1}{3})$

  hard $\in [\frac{1}{3}, \frac{1}{2})$ very hard $\in [\frac{1}{2}, 1)$

Relative Frequency

- Is easily controlled; simply select the desired number of anomalies. In this study we examined relative frequencies of 0.001, 0.005, 0.01, 0.05, and 0.1

Semantic Variation

- We define *clusteredness* to indicate the semantic variation in the chosen anomalies as the ratio of the sample variance of the nominal points over the sample variance of the anomalous points.
- We partially control this problem dimension by using a facility location algorithm to choose clustered or scattered groups.
- We bucket the scores into six qualitative groups:

high scatter $\in (0, 0.25)$	medium scatter $\in [0.25, 0.5)$	low scatter $\in [0.5, 1)$
low cluster $\in [1, 2)$	medium cluster $\in [2, 4)$	high cluster $\in [4, \infty)$

Algorithm Summary

- One-Class SVM (ocsvm) (Scholkopf et al.)
- Support Vector Data Description (svdd) (Tax and Duin)
- Local Outlier Factor (lof) (Breunig et al.)
- Isolation Forest (if) (Liu et. al)
- (New) Robust Kernel Density Estimation (rkde) (Kim and Scott)
  - J. Kim and C. Scott, Robust Kernel Density Estimation. *Journal of Machine Learning Research*, 13 (2012) 2529-2565
- Ensemble Gaussian Mixture Model (egmm)

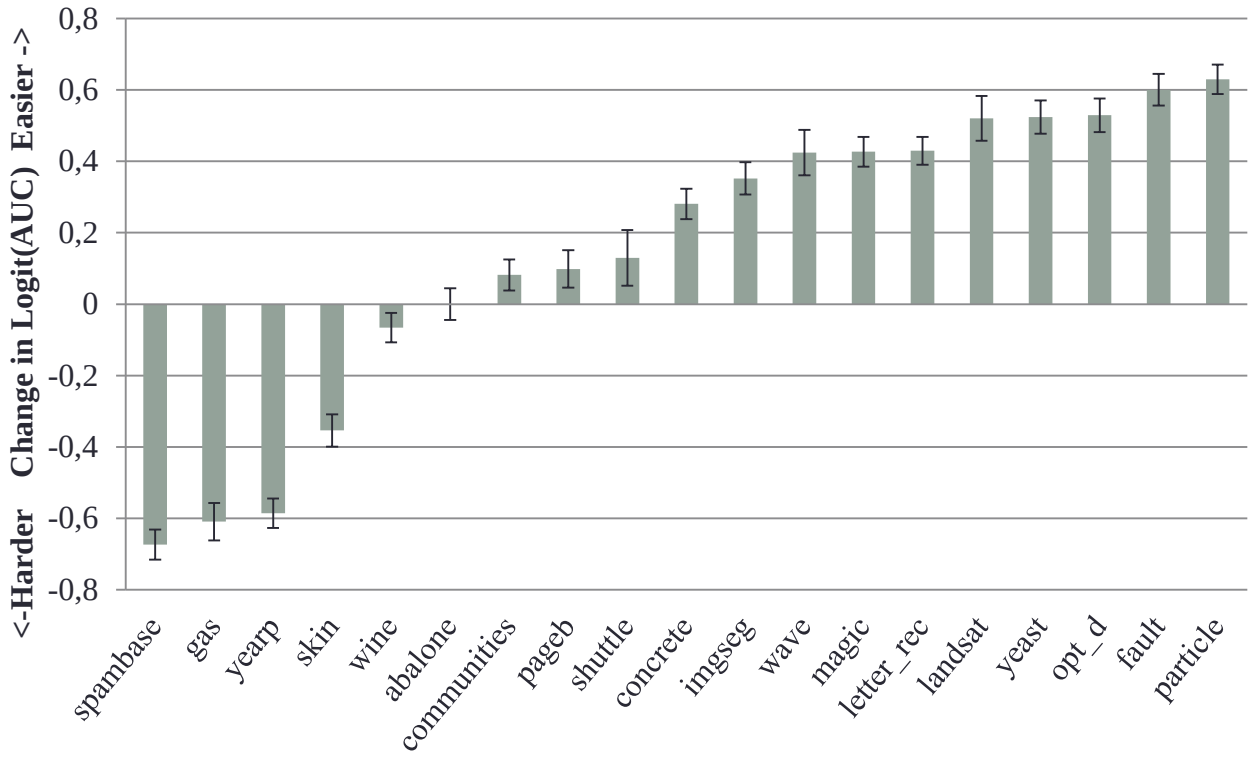
Results

- We evaluate each algorithm on each benchmark with ROC AUC.

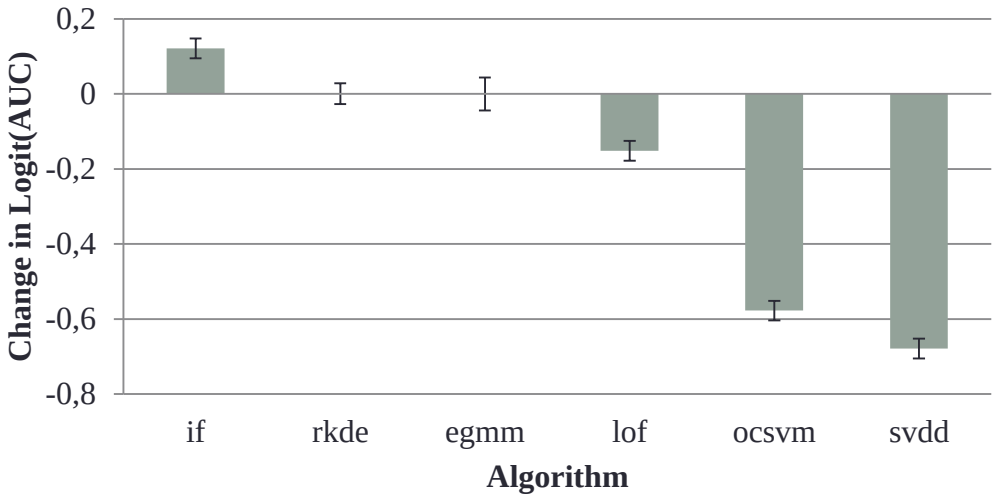
- We fit a linear model and measure Δ logit(AUC)

$$logit(AUC) \sim set + algo + diff + freq + cluster$$

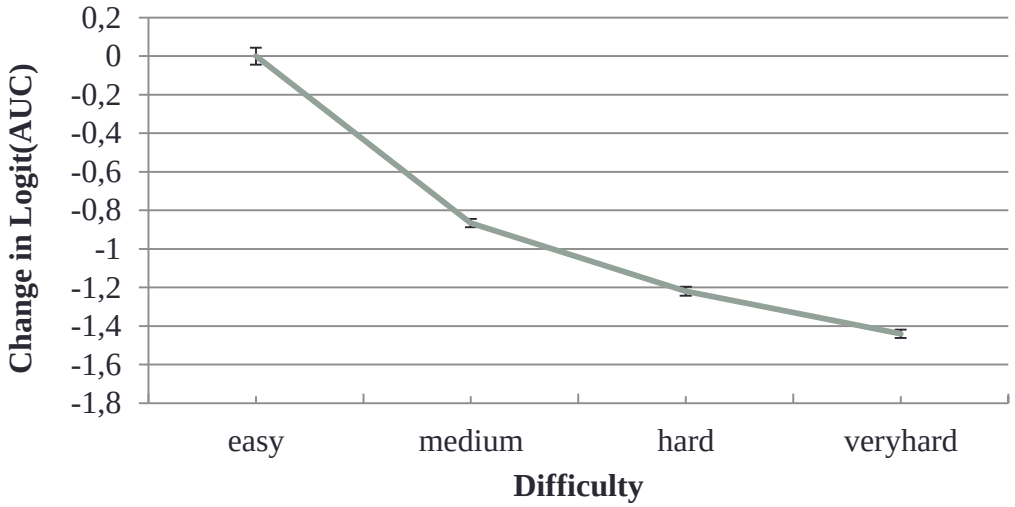
Δ logit(AUC) by set



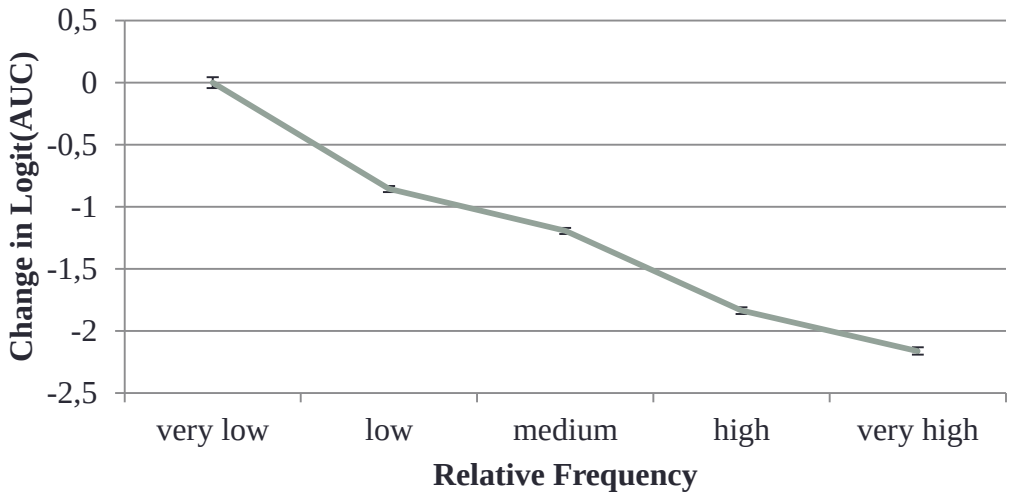
Δ logit(AUC) by algorithm



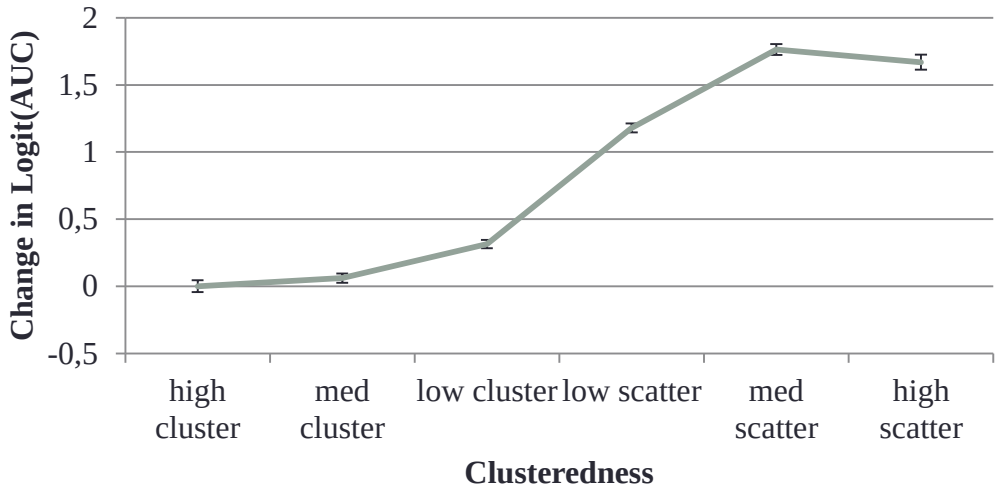
Δ logit(AUC) by difficulty



Δ logit(AUC) by relative frequency



Δ logit(AUC) by clusteredness



Thank You!