On the Internal Evaluation of Unsupervised Outlier Detection

Henrique O. Marques[1], Ricardo J. G. B. Campello[1],
Arthur Zimek[2], Jörg Sander[3]

[1]University of São Paulo, São Carlos, SP, Brazil

[2]Ludwig-Maximilians-Universität München, Munich, Germany

[3]University of Alberta, Edmonton, AB, Canada

SSDBM 2015, San Diego, CA

Motivation

Detecting patterns that are exceptional in some sense is relevant:

- Removal of spurious observations prior to data analysis
  - noise, sensor failures, ...
- Extraordinary behaviors that deserve some special attention
  - genes associated with certain diseases
  - frauds in financial systems
  - employees with unusual productivity profiles
  - ...

Outlier detection techniques

Techniques can be categorized in different ways:

- Supervised, semi-supervised or unsupervised
- Binary (top-n) vs ranking/scoring-based

Outlier detection techniques

Techniques can be categorized in different ways:

- Supervised, semi-supervised or unsupervised
- Binary (top-n) vs ranking/scoring-based

What is an outlier?

In the unsupervised scenario, it is not precisely defined

Different definitions try to capture the same intuitive idea

Outlier

"An observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism"

Hawkins

Evaluation in unsupervised learning

- Data clustering
  - Internal indexes (e.g. silhouette) have been extensively used
    - for model selection
    - for statistical validation
- Outlier detection
  - The internal evaluation problem has been surprisingly overlooked in unsupervised outlier detection

How do people evaluate their outlier detection results then?

In the literature: mostly restricted to controlled experiments using *external* evaluation measures, i.e., based on a *ground truth*

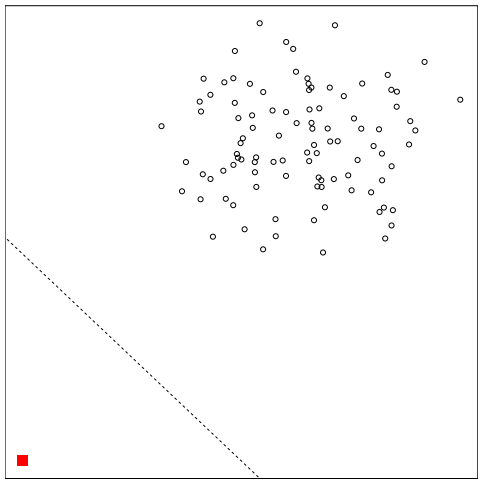
- Precision-at-n (prec@n)
- AUC ROC

In practice (*no ground truth is available*): we are not aware of the existence of any internal evaluation index

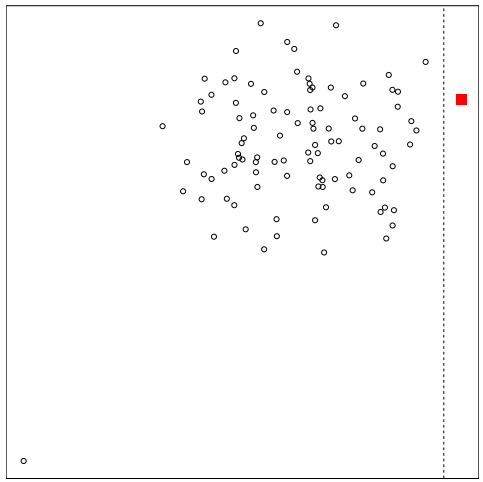
Problem statement

- $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$: an unlabeled data set with N objects
- $\mathbf{S} \subset \mathbf{X}, |\mathbf{S}| = n$: a binary (top-$n$) outlier detection solution
- Given a collection of such candidate solutions, we want to independently quantify the quality of each individual solution:
  - to assess their statistical significance against random solutions
  - to compare them in relative terms
    - best candidates $\Leftrightarrow$ more suitable algorithms / parameters

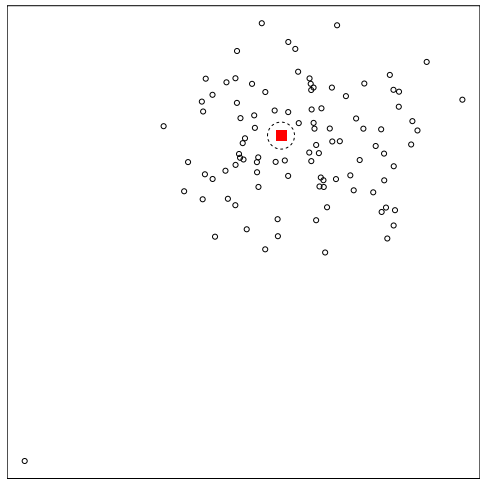
Basic intuition



(a) Global outlier

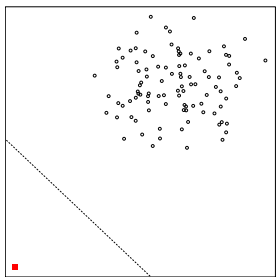


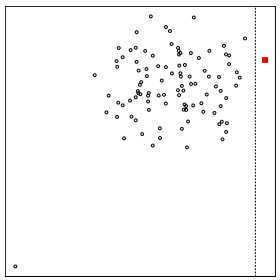
(b) Local outlier

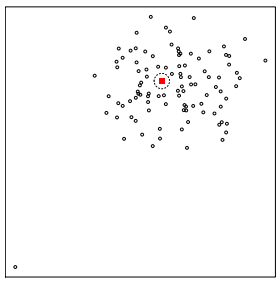


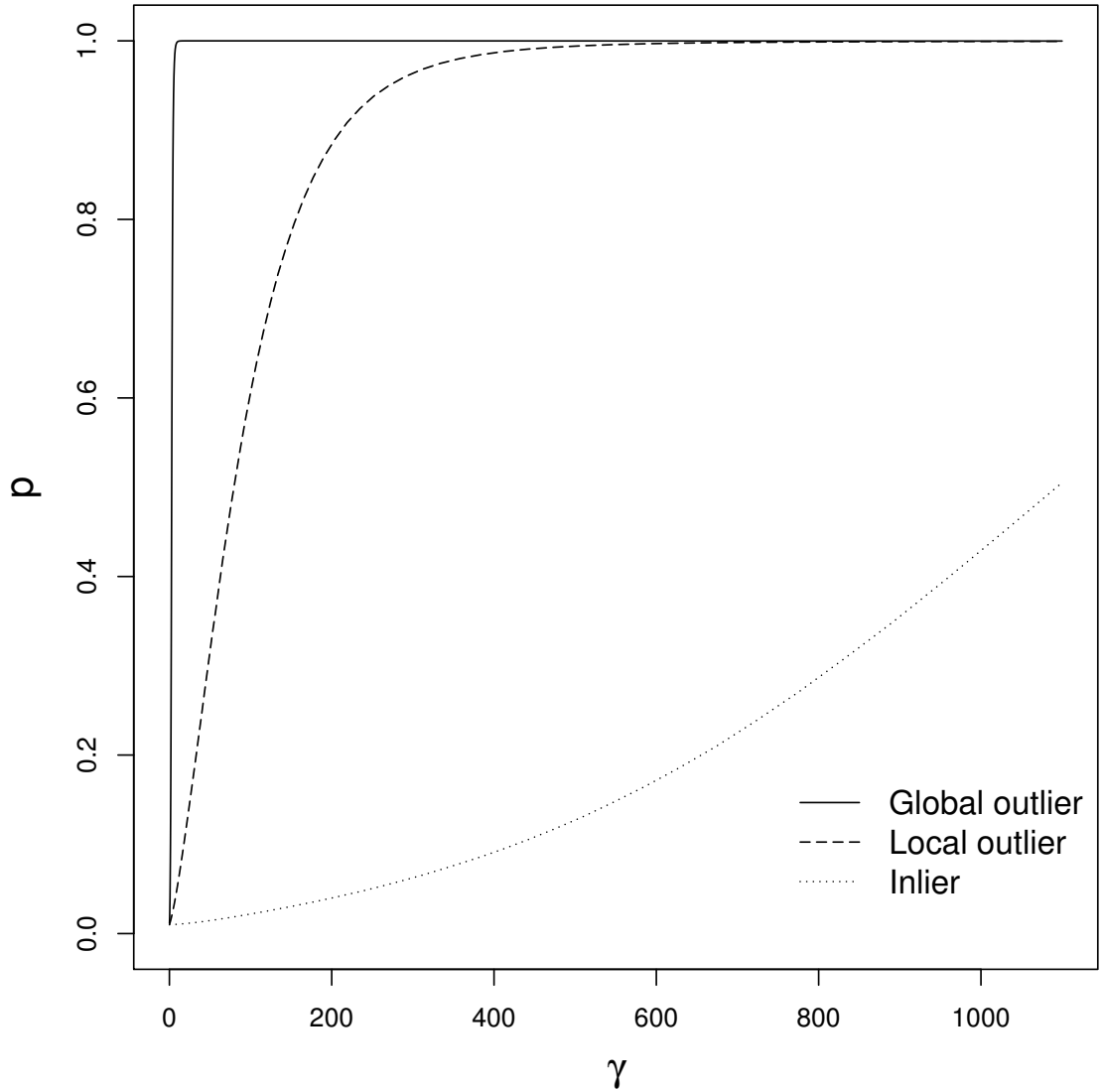
(c) Inlier

Separability curve (max. margin classifier)









A baseline index

Given a top-n outlier detection solution $\mathbf{S}$:

$$I(\mathbf{S}) = \frac{1}{\gamma_{max}} \int_{\gamma=0}^{\gamma_{max}} \bar{p}(\gamma) \qquad (1)$$

- $\bar{p}(\gamma)$: separability averaged over all objects $\mathbf{x}_j \in \mathbf{S}$

A baseline index (practical)

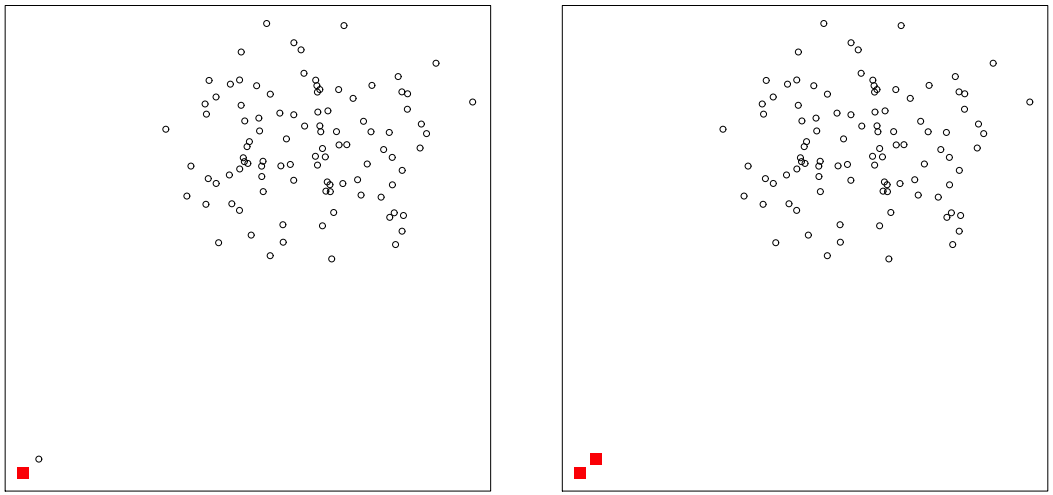
Given a top-n outlier detection solution $\mathbf{S}$:

$$I(\mathbf{S}) \approx \frac{1}{n_\gamma} \sum_{l=1}^{n_\gamma} \bar{p}(\gamma_l) \tag{2}$$

- $\bar{p}(\gamma)$: separability averaged over all objects $\mathbf{x}_j \in \mathbf{S}$

Intuitions missing in the baseline index

1. Maximum clump size, m_{cl} (optional):
   - What is judged to be more likely a clump of potential outliers
     - background noise lumps, outlierish micro-clusters, ...
2. The negative impact of nearby objects should be more severe if they are assigned a different label



Incorporating the missing intuitions

Soft margin classifiers (penalty P for margin violations):

$$P = \sum_{i=1}^{N} C_i g(\mathbf{x}_i) \tag{3}$$

$C_i = C \rightarrow$ inlier (full cost)
$C_i = \beta C \rightarrow$ outlier (fractional cost)
$\beta = 1/m_{cl}$

IREOS index

$$I(\mathbf{S}) = \frac{1}{n_\gamma} \sum_{l=1}^{n_\gamma} \bar{p}(\gamma_l) \tag{4}$$

- Not hooked on any specific soft margin classifier
- We used Kernel Logistic Regression (KLR) in our experiments
  - Automatically provides the probability p that each object $\mathbf{x}_j$ belongs to the outlier class (separability as a byproduct)
    - Separability naturally normalized (as probabilities) within $[0, 1]$
  - KLR is known to be robust even in the presence of imbalanced classes and small amounts of training data

Adjustment for Chance and Statistical Validation

Central Limit Theorem (CLT): the sample mean $\bar{p}(\gamma)$ follows at least approximately a Normal distribution

$$\bar{p}(\gamma) \sim \mathcal{N}(E\{\bar{p}(\gamma)\}, Var\{\bar{p}(\gamma)\}) \tag{5}$$

IREOS is given by a sum of normally distributed variables, so:

$$I \sim \mathcal{N}(E\{I\}, Var\{I\}) \tag{6}$$

If we know the mean and variance, we can:

- Adjust IREOS for chance
- Assess the statistical significance of a solution (e.g. z-test)

Adjustment for Chance

$$I_{adj}(\mathbf{S}) = \frac{I(\mathbf{S}) - E\{I\}}{I_{max} - E\{I\}} \tag{7}$$

$$E\{I\} = \frac{1}{n_\gamma} \sum_{l=1}^{n_\gamma} E\{\bar{p}(\gamma_l)\} \tag{8}$$

$$E\{\bar{p}(\gamma_l)\} = \frac{1}{N} \sum_{\mathbf{x}_j \in \mathbf{X}} p(\mathbf{x}_j, \gamma_l) \tag{9}$$

Statistical Validation

$$Var\{I\} = \frac{1}{n_\gamma^2} \sum_{l=1}^{n_\gamma} Var\left\{\bar{p}(\gamma_l)\right\} + \frac{2}{n_\gamma^2} \sum_{l_1=1}^{l_2-1} \sum_{l_2=2}^{n_\gamma} Cov(\bar{p}(\gamma_{l_1}), \bar{p}(\gamma_{l_2})) \tag{10}$$

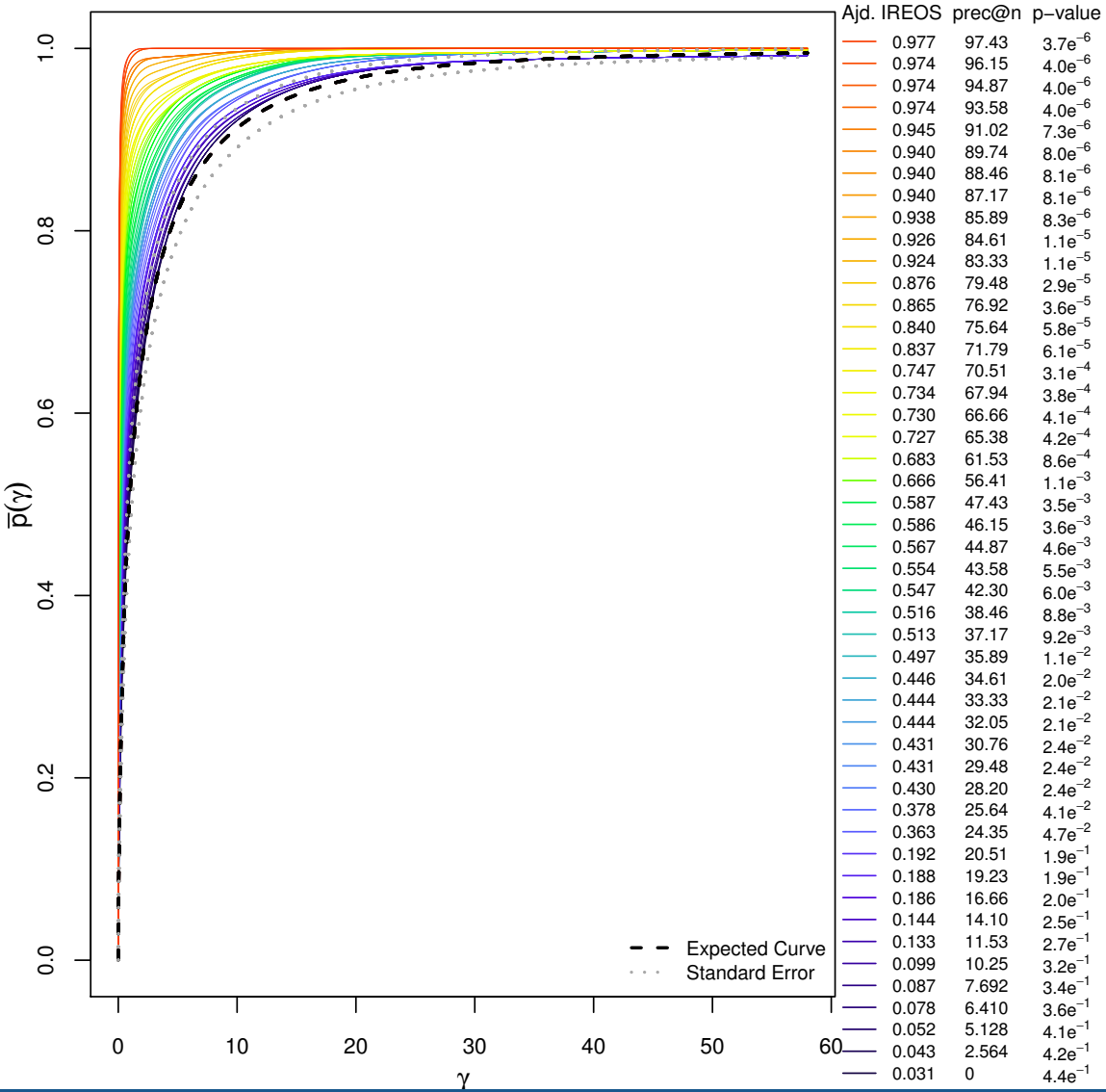
$$Var\{\bar{p}(\gamma_l)\} = \frac{1}{n} Var\{p(\mathbf{x}_j, \gamma_l)\} \tag{11}$$

$$Cov(\bar{p}(\gamma_{l_1}), \bar{p}(\gamma_{l_2})) = \frac{1}{n} Cov(p(\mathbf{x}_j, \gamma_{l_1}), p(\mathbf{x}_j, \gamma_{l_2})) \tag{12}$$

Approximate Computation via Monte Carlo

- Exact computations presume $m_{cl} = 1$ (clumps not modeled)
- **Monte Carlo** simulations can be used to estimate statistics rather than try to compute them in an exhaustive way
- The Monte Carlo sample size represents a trade-off between computational burden and accuracy

First type experiments: controlled experiment

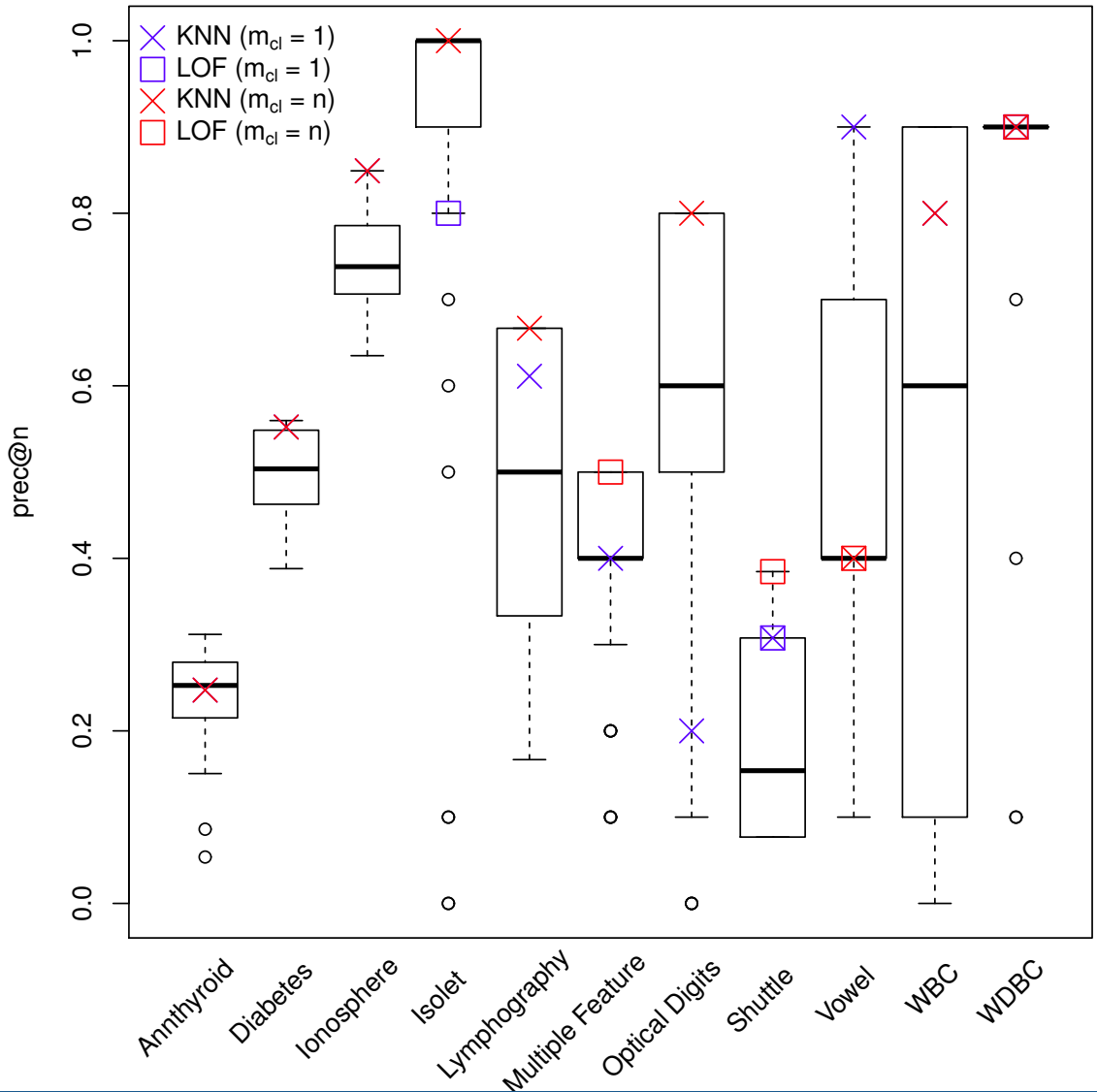


First type experiments: controlled experiment

Dataset	$m_{cl} = 1$	$m_{cl} = n$
Zimek's data collection	0.995 ± 0.011	0.996 ± 0.012
Handl's data collection	0.998 ± 0.004	0.994 ± 0.02
Annthyroid	0.999	0.999
Diabetes	0.997	0.64
Ionosphere	0.998	0.948
Isolet	1	1
Lymphography	1	1
Multiple Features	0.981	0.99
Optical Digits	1	1
Shuttle	0.52	0.995
Vowel	1	1
WBC	1	
WDBC	1	1

Table: Spearman correlation between IREOS and prec@n: synthetic data collections (top two) and real datasets (bottom)

Second type experiments: model selection



Second type experiments: model selection

Dataset	Min	Max	Avg	IREOS ($m_{cl} = 1$)	IREOS ($m_{cl} = n$)
Annthyroid	0.0538	0.3118	0.241	0.2473	0.2473
Diabetes	0.3881	0.5597	0.4966	0.5522	0.5522
Ionosphere	0.6349	0.8492	0.7386	0.8492	0.8492
Isolet	0	1	0.8353	0.8	1
Lymphography	0.1667	0.6667	0.4606	0.6111	0.6667
Multiple Features	0.1	0.5	0.3882	0.4	0.5
Optical Digits	0	0.8	0.5765	0.2	0.8
Shuttle	0.0769	0.3846	0.1855	0.3077	0.3846
Vowel	0.1	0.9	0.5324	0.9	0.4
WBC	0	0.9	0.5265	0.8	0.8
WDBC	0.1	0.9	0.8324	0.9	0.9

Table: prec@n for LOF and kNN outlier solutions with varied parameters (neighborhood size)

Robustness to the penalty (soft margin) cost

Cost C	$m_{cl} = 1$	$m_{cl} = n$
100	0.996 ± 0.009	0.997 ± 0.008
1000	0.998 ± 0.004	0.994 ± 0.02
20000	0.998 ± 0.001	0.995 ± 0.01
800000	0.997 ± 0.003	0.993 ± 0.018

Table: Spearman correlation between IREOS and prec@n for varied cost values C (Handl's data collection)

Monte Carlo simulations

	$E\{I\}$	Estimated $E\{I\}$	Worst Abs. Difference
1%	0.941	0.940 ± 0.023	0.068
2%	0.941	0.941 ± 0.014	0.044
5%	0.941	0.942 ± 0.007	0.018
10%	0.941	0.941 ± 0.006	0.012
20%	0.941	0.940 ± 0.004	0.01

Table: 30 runs for varied sample sizes n_{MC} corresponding to different percentages of the population

Conclusion

- IREOS (Internal, Relative Evaluation of Outlier Solutions): quantitative, unsupervised evaluation of top-n outliers
- Adjustment for chance and statistical validation
- Experiments with synthetic and real data display high correlation between IREOS and the ground truth

Ongoing work

- How to compare solutions with different values of n
  - thus being able to automatically determine an optimal n
- How to internally evaluate non-binary solutions
  - thus being able to compare rankings/scorings of outliers

Thank you for your attention!