A comparison of AUC-estimators in small-sample studies

Antti Airola Tapio Pahikkala
Willem Waegeman Bernard De Baets
Tapio Salakoski

Turku Centre for Computer Science (TUCS)
University of Turku, Department of Information Technology
University of Ghent, KERMIT, Department of Applied Mathematics, Biometrics and Process Control

September 6, 2009

Overview

- AUC (Area under ROC curve): classification performance measure
- Cross-validation typically used to measure AUC when data is scarce
- But how to do it right?
- Pooled vs. averaged?
- Tenfold vs. leave-one-out vs. leave-pair-out?
- We explore this through a simulation study

Presentation outline

1. Preliminaries
2. Cross-validation
3. Simulation study

Binary classification

- Input: A training set $Z = ((x_1, y_1) \dots (x_m, y_m))$ of m (attributes, label) pairs sampled from a probability distribution D
- Possible labels are $\{-1, +1\}$, that is, each example belongs either to the "negative" or to the "positive" class
- Task: To learn, a prediction function f_Z, which is able to predict the label y' given the attributes x' of a new example drawn from D
- Assumption: f_Z real-valued

Measuring the performance of a classifier

AUC

- Area under receiver operating characteristic curve
- Ranking based measure of classification performance
- Probability, that a randomly chosen positive example receives higher predicted value than a randomly chosen negative one
- Insensitive to relative class distributions and class-specific error costs
- Popular in machine learning, medical decision making, microarray studies...

Conditional performance

Conditional expected AUC:

$$A(f_Z) = E_{x_+ \sim D_+ x_- \sim D_-}[\delta(f_Z(x_+) - f_Z(x_-))]$$

$$\delta(a) = \begin{cases} 1 & \text{when } a > 0 \\ 1/2 & \text{when } a = 0 \\ 0 & \text{when } a < 0 \end{cases}$$

- assumes a fixed training set Z, from which we learn f_Z
- measures the generalization performance of f_Z

Measuring estimator quality

We can almost never directly calculate $A(f_Z)$, use some estimate $\hat{A}(f_Z)$ instead

- deviation $\hat{A}(f_Z) - A(f_Z)$
- $E_{Z \sim D^m}[\hat{A}(f_Z) - A(f_Z)]$ (bias)
- $Var_{Z \sim D^m}[\hat{A}(f_Z) - A(f_Z)]$ (variance)

Unconditional performance

Unconditional expected AUC:

$$E_{Z \sim D^m}[A(f_Z)].$$

- considering all possible training sets (of a fixed size)
- how good prediction function f_Z will our learning method on average give us?
- In machine learning literature focus usually on measuring the quality of learning algorithms, training data treated as a random variable
- However, conditional performance in many cases of more practical interest
- Instead of the average case we want to know how good a prediction function we can learn from our particular dataset

Estimating conditional performance

Wilcoxon-Mann-Whitney statistic

$$\hat{A}(S, f_Z) = \frac{1}{|S_+||S_-|} \sum_{x_i \in S_+} \sum_{x_j \in S_-} \delta(f_Z(x_i) - f_Z(x_j))$$

S: a sequence of examples
$S_+ \subset S$ and $S_- \subset S$ the positive and negative examples in S.

- How should we choose S?
- Training set performance unreliable due to overfitting
- Separate test set cannot be afforded for small datasets
- Cross-validation

Cross-validation

- $\mathcal{H} = \{H_1, \ldots, H_N\}$: a sequence of hold-out sets
- On each cross-validation round, learn $f_{\overline{H_i}}$ from non-holdout examples, and predict on holdout examples
- Fold-wise predictions from cross-validation $\{\hat{Y}_{H_1} \ldots \hat{Y}_{H_N}\}$
- Corresponding correct labels $\{Y_{H_1} \ldots Y_{H_N}\}$

- Two approaches to AUC estimation
- Averaging: Calculate AUC separately for each $(\hat{Y}_{H_i}, Y_{H_i})$-pair and sum these together
- Pooling: Calculate one global AUC estimate over the pair $(\hat{Y}_{H_1} \cup \ldots \cup \hat{Y}_{H_N}, Y_{H_1} \cup \ldots \cup Y_{H_N})$

Averaged AUC Performance

$$N \sum_{H \in \mathcal{H}} \sum_{i \in H_+, j \in H_-} \delta(f_{\overline{H}}(x_i) - f_{\overline{H}}(x_j))$$

Notation:

$$\begin{aligned}
\mathcal{H} &= \text{Set of hold-out sets} \\
H &= \text{hold-out set} \\
H_+ &= \text{indices of the positive examples in the hold-out set} \\
H_- &= \text{indices of the negative examples in the hold-out set} \\
\overline{H} &= \text{complement of the hold-out set} \\
f_{\overline{H}} &= \text{the learning method trained with examples belonging to } \overline{H} \\
N &= \text{normalizing constant}
\end{aligned}$$

Pooled AUC Performance

$$N \sum_{H,H' \in \mathcal{H}} \sum_{i \in H_+, j \in H'_-} \delta(f_{\overline{H}}(x_i) - f_{\overline{H'}}(x_j))$$

Notation:

$$\begin{aligned}
\mathcal{H} &= \text{Set of hold-out sets} \\
H &= \text{hold-out set} \\
H_+ &= \text{indices of the positive examples in the hold-out set} \\
H_- &= \text{indices of the negative examples in the hold-out set} \\
\overline{H} &= \text{complement of the hold-out set} \\
f_{\overline{H}} &= \text{the learning method trained with examples belonging to } \overline{H} \\
N &= \text{normalizing constant}
\end{aligned}$$

Leave-pair-out cross-validation

The set of hold-out sets consists of each possible pair of positive-negative training example pairs.

$$\frac{1}{m_+ m_-} \sum_{\{i,j\} \in \mathcal{H}} \delta(f_{\overline{\{i,j\}}}(x_i) - f_{\overline{\{i,j\}}}(x_j))$$

Notation:

m_+ = the number of training examples in the positive class

m_- = the number of training examples in the negative class

$f_{\overline{\{i,j\}}}$ = classifier trained without the i-th and j-th training example

Different cross-validation strategies

N-fold cross-validation

- split data into N mutually disjoint folds
- 10-fold most commonly used
- possible to use both averaging and pooling

Leave-one-out

- each example held out in turn
- averaging not possible, only pooling

Leave-pair-out

- each positive-negative example pair held out in turn
- natural for AUC, which is defined over all positive-negative pairs

Simulation study

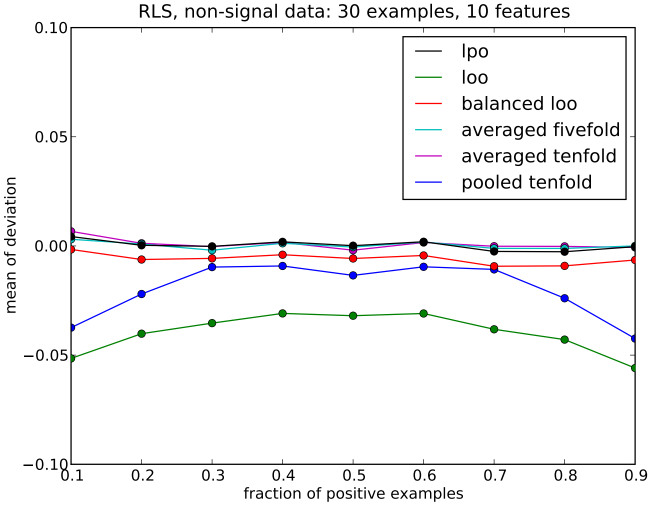
- Compare several different cross-validation strategies
- high- and low dimensional, signal- and non-signal data
- 10000 repetitions of each experiment, training sets of 30 examples, test sets of 10000 examples
- Deviation $\hat{A}(f_Z) - A(f_Z)$ as a measure of quality of $\hat{A}$
- Mean and variance of deviation
- RLS and RankRLS, linear kernel

Simulation study

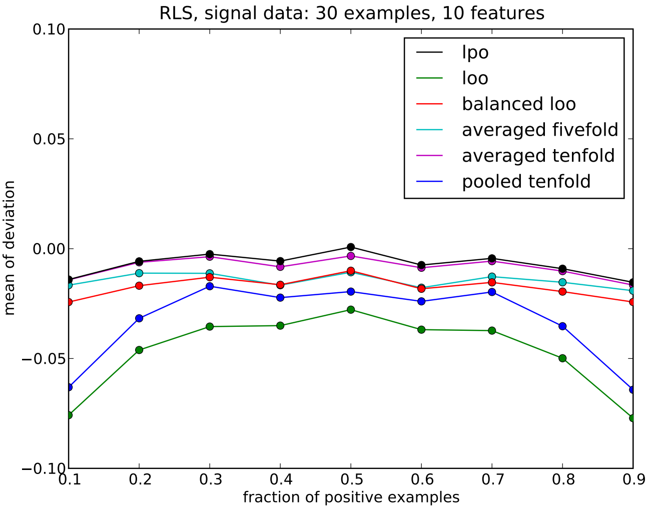
Compared methods:

- leave-one-out (pooled)
- balanced leave-one-out (pooled) (Parker et al. 2007)
- leave-pair-out (averaged)
- averaged fivefold
- pooled tenfold
- averaged tenfold

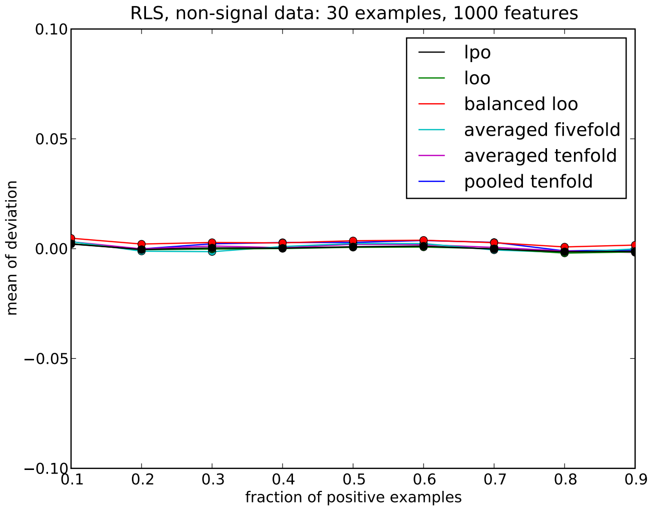
Simulation study



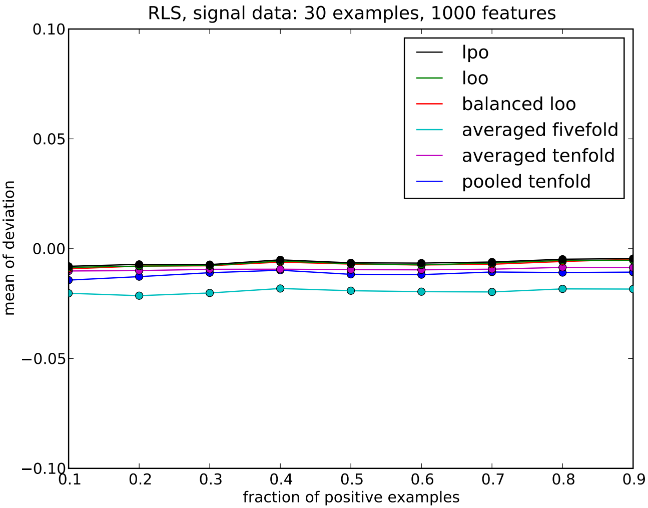
Simulation study



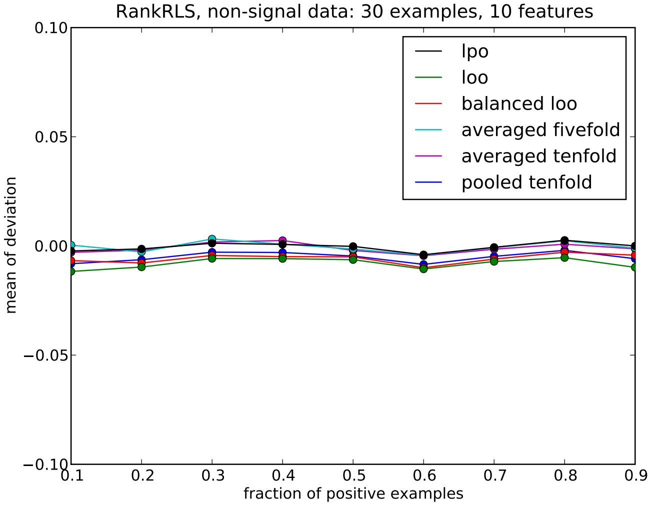
Simulation study



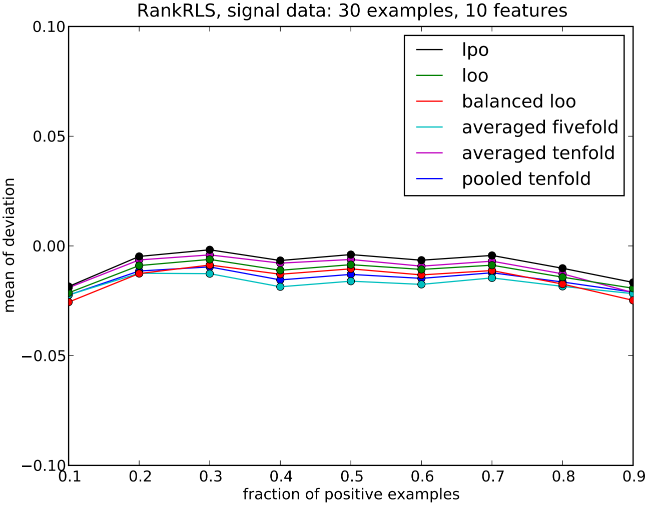
Simulation study



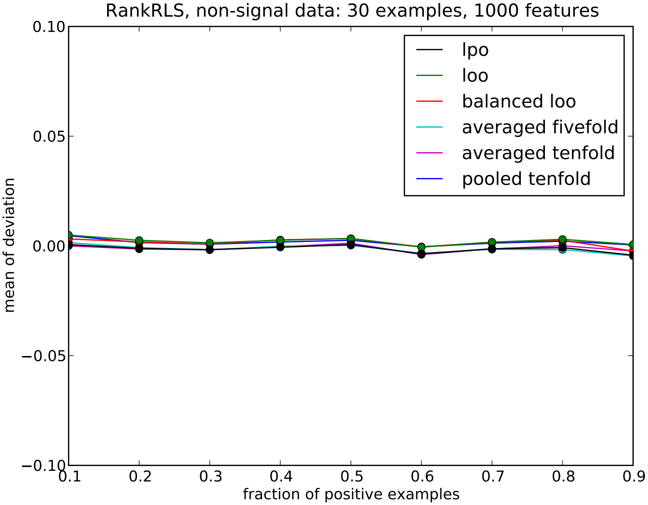
Simulation study



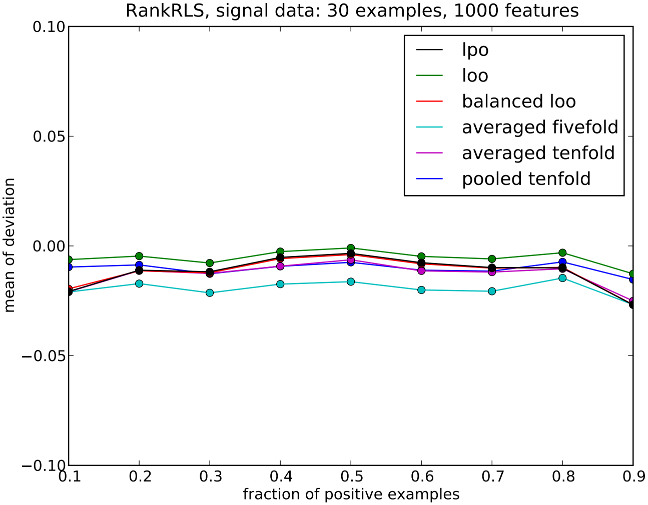
Simulation study



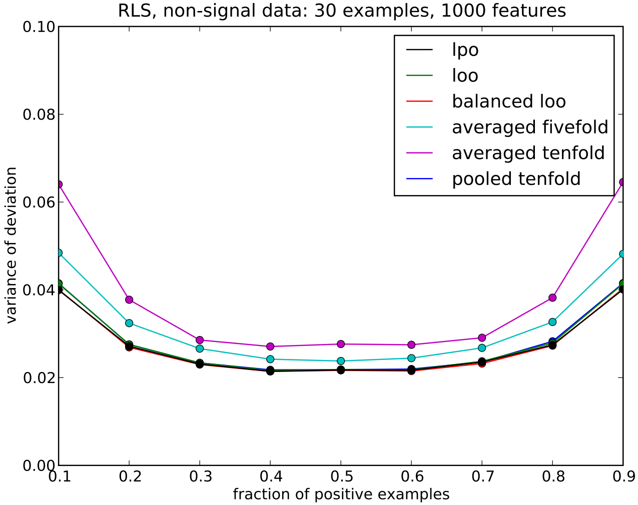
Simulation study



Simulation study



Simulation study



Conclusion

Main findings:

- Pooled estimators negatively biased on low dimensional data
- Averaged tenfold and fivefold have high variance
- LPOCV: almost unbiased and competitive variance

Recommendations:

- LPOCV most reliable, if it can be afforded
- Pooling also reliable on high dimensional data?

RLScore: `www.tucs.fi/rlscore`