A comparison of AUC estimators in small-sample studies

Antti Airola,[1] Tapio Pahikkala,[1] Willem Waegeman,[2] Bernard De Baets,[2] and Tapio Salakoski[1]

[1] Department of Information Technology, University of Turku and Turku Centre for Computer Science (TUCS), Joukahaisenkatu 3-5 B, Turku, Finland
[2] KERMIT, Department of Applied Mathematics, Biometrics and Process Control, Coupure links 653, Ghent University, Belgium

Abstract. Reliable estimation of the classification performance of learned predictive models is difficult, when working in the small sample setting. When dealing with biological data it is often the case that separate test data cannot be afforded. Cross-validation is in this case a typical strategy for estimating the performance. Recent results, further supported by experimental evidence presented in this article, show that many standard approaches to cross-validation suffer from extensive bias or variance when the area under ROC curve (AUC) is used as performance measure. We advocate the use of leave-pair-out cross-validation (LPOCV) for performance estimation, as it avoids many of these problems.

1 Introduction

Small-sample biological datasets, such as microarray data, exhibit properties which pose serious challenges for reliable evaluation of the quality of prediction functions learned from this data. It is typical for genomic studies to produce data containing thousands of features, measured from a small sample of possibly only tens of examples. Further, the relative distribution of the classes to be predicted is often highly imbalanced and their discriminability can be quite low.

AUC is a ranking-based measure of classification performance, which has gained substantial popularity in the machine learning community during recent years [1–3]. Its value can be interpreted as the probability that a classifier is able to distinguish a randomly chosen positive example from a randomly chosen negative example. In contrast to many alternative performance measures, AUC is invariant to relative class distributions, and class-specific error costs. These properties have prompted the use of the AUC measure in microarray studies [4, 5], medical decision making [6], and evaluation of biomedical text mining systems [7] to name a few examples.

When setting aside data for parameter estimation and validation of results cannot be afforded, cross-validation is typically used. However, in [8] it was shown that when considering AUC in the small-sample setting, many commonly used cross-validation schemes suffer from substantial negative bias. In this work, we

explore this issue further and propose LPOCV, first considered in [9] for ranking tasks, as an approach that provides an almost unbiased estimate of expected AUC performance, and also does not suffer from as high variance as some of the alternative strategies.

2 Performance Estimation

Let D be a probability distribution over a sample space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, where the input space $\mathcal{X}$ is a set and the output space $\mathcal{Y} = \{-1, 1\}$. An example $z = (x, y) \in \mathcal{Z}$ is thus a pair consisting of an input and an associated label, which describes whether the example belongs to the positive or to the negative class. The conditional distribution of an input from $\mathcal{X}$, given that it belongs to the positive class is denoted by D_+, and given that it belongs to the negative class by D_-. Further, let the sequence $Z = ((x_1, y_1), \ldots, (x_m, y_m)) \in \mathcal{Z}^m$ drawn independent and identically distributed from D be a training set of m training examples, with $X = (x_1, \ldots, x_m) \in \mathcal{X}^m$ denoting the inputs and $Y = (y_1, \ldots, y_m) \in \mathcal{Y}^m$ the labels in the training set.

Now let us consider a prediction function f_Z returned by a learning algorithm based on a fixed training set Z. We are interested in the generalization performance of this function, that is, how well it will predict on unseen future data. The generalization performance of f_Z can be measured by its expected AUC $A(f_Z)$, sometimes also known as expected ranking accuracy [10], over all possible positive-negative example pairs, that is

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

where H is the Heaviside step function, for which $H(a)$ is 1 if $a > 0$, 1/2 if $a = 0$, and 0 if $a < 0$. We call this measure the *conditional expected AUC* of the prediction function, as it is conditioned on a fixed training set Z.

Alternatively, we may also want to consider the expectation taken over all possible training sets of size m. The *unconditional expected AUC* can be defined as

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

As discussed for example in [11, 12], these two measures correspond to two different questions of interest. The conditional expected performance corresponds to the question how well we expect that a prediction function learned from a given training set will generalize to future examples. The unconditional expected performance measures the quality of the learning algorithm itself, that is, how well on average will a prediction function learned by the algorithm of interest from a dataset of a given size generalize to new data.

More often, machine learning related articles concentrate on the unconditional performance, as the goal usually is to measure the quality of learning algorithms, where the training data is treated as a random variable. However, as argued by [11], the conditional error estimate is more of interest in a setting

where a researcher is using a certain dataset and wants to know how well a prediction function learned from that particular dataset will do on future examples. This is the setting we concentrate on in this paper.

In practice we almost never can directly access the probability distribution D to calculate A, but are rather limited to using some estimate $\hat{A}$ instead. To measure the quality of an estimator, in terms of its ability to measure conditional expected AUC, we follow the setting of [11]. We consider the deviation $B(Z) = \hat{A}(f_Z) - A(f_Z)$, which measures the difference between the estimated and true conditional expected AUC of a prediction function.

We study the expected value $E_{Z\sim D^m}[B(Z)]$ of the deviation distribution as a measure of the biasedness of the estimator. Further, we consider the variance $\mathrm{Var}_{Z\sim D^m}[B(Z)]$ of the deviation distribution, as a measure of the reliability of individual estimates. Preferably an estimator would have both close to zero deviation mean and variance.

The AUC measure can be calculated using the following formula, also called the Wilcoxon-Mann-Whitney statistic:

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

where S is a sequence of examples, and $S_+ \subset S$ and $S_- \subset S$ denote the positive and negative examples in S, respectively. (for proof, see [13]).

In this paper, we consider a commonly used performance evaluation technique known as cross-validation. Here, the dataset is repeatedly partitioned into two non-overlapping parts, a training set and a hold-out set. For each partitioning, the hold-out set is used for testing while the remainder is used for training. The two most popular variants are *tenfold cross-validation*, where the data is split into ten mutually disjoint folds, and *leave-one-out cross-validation* (LOOCV), where each training example constitutes its own fold.

Stratification is commonly done to ensure that the hold-out sets share approximately the same class distributions. Further, for stratified CV on small datasets [8] has recently suggested a balancing strategy to ensure that all the training sets share the same number of positive and negative examples. When the sample size for a class is not a multiple of the number of folds, some folds will contain one extra example from that class compared to the other folds. The balancing is done by randomly removing members of overrepresented classes on each round of cross-validation, so that all the training sets contain the same number of examples from each class.

As discussed in [1, 8], two alternative strategies can be used to calculate the cross-validation estimate over the folds, *pooling* and *averaging*.

In pooling, the predictions made in each cross-validation round are pooled into a one set and one common AUC score is calculated from it. For LOOCV this is the only way to obtain the AUC score. The assumption made when using pooling is that classifiers produced on different cross-validation rounds come from the same population. This assumption may make sense when using performance measures such as classification accuracy, but it is more dubious

when computing AUC, since some of the positive-negative pairs are constructed using data instances from different folds. Indeed, [8] show that this assumption is generally not valid for cross-validation and can lead to large pessimistic biases. In their experiments with no-signal data sets, AUC values of less than 0.3 were observed instead of the expected 0.5.

An alternative approach, averaging, is to calculate the AUC score separately for each cross-validation fold and average them to obtain one common performance estimate. However, the number of positive-negative example pairs in the folds may be too small for calculating AUC reliably when using small imbalanced datasets. As an extreme case, if there are more folds than observations for the minority class, then some of the folds cannot have examples from this class. For such folds, the AUC cannot be calculated.

LPOCV [9, 14] was first introduced for general ranking tasks. Here, we propose its use for AUC calculation, since it avoids many of the pitfalls associated with the pooling and averaging techniques. Analogously to LOOCV, each possible positive-negative pair of training instances is left out of at a time from the training set. Formally, the AUC performance is calculated with LPOCV as

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

where $f_{\overline{\{i,j\}}}$ denotes a classifier trained without the i-th and j-th training example. Being an extreme form of averaging, where each positive-negative pair of training examples forms an individual hold-out set, this approach is natural when AUC is used as a performance measure, since it guarantees the maximal use of available training data. Moreover, the LPOCV estimate, taken over a training set of m examples, is an unbiased estimate of the unconditional expected AUC over a sample of $m-2$ examples (for a proof, see [9]).

The computational cost can be seen as a limitation for cross-validation techniques in general, and in particular for the LOOCV and LPOCV. For a training set of m examples a straightforward implementation of LOOCV requires training the learner m times, with LPOCV the required number of training rounds is of the order $O(m^2)$. While these computational costs may be affordable on small training sets, they can become a limiting factor as the training set size increases.

However, for regularized least-squares (RLS) [15] and the AUC-maximizing ranking RLS (RankRLS) [16], efficient algorithms for cross-validation can be derived using techniques based on matrix calculus [17, 14]. Since these algorithms have state-of-the-art classification performance similar to that of the Support Vector Machine (SVM), and Ranking SVM (see e.g. [18, 16]), they are a natural choice to use in settings where cross-validation is important.

3 Empirical study

In the simulation study, we measure the mean and variance of the deviation distribution of several different cross-validation estimators. We consider three

pooled strategies; LOOCV, balanced LOOCV and pooled tenfold, as well as the averaged fivefold, tenfold and LPOCV. Stratification is used where possible.

Our setting is similar to that of [8], where the bias of pooling and averaging approaches was compared on low-dimensional data. We consider synthetic data, as this allows estimating the conditional expected AUC of the learned prediction functions. The training set size is 30 examples in all the simulations, the relative distribution of positive examples is varied between 10% and 90% on 10 percentage unit intervals. We consider both low-dimensional data with 10 features, and high-dimensional data with 1000 features.

In the no-signal experiment, there is no difference between the two classes. Examples from both classes are sampled from normal distributions with zero mean, unit variance and no covariance between the features. The conditional expected AUC of a prediction function is in this setting 0.5, as no model can do either better or worse than random, in terms of AUC. In the signal experiment the means of a number of features are shifted to 0.5 for the positive, and to -0.5 for the negative class. With 10 features, 1 feature is shifted, with 1000 features, 10 features are shifted. Generated test sets with 10000 examples are used to estimate the conditional expected AUC of the learned prediction functions.

Two learning algorithms are considered in the experiments, RLS and RankRLS. RLS optimizes an approximation of accuracy, like most machine learning algorithms, while RankRLS optimizes more directly the AUC. We only investigated the linear kernel, since in bioinformatics it is commonly assumed that high-dimensional data can be separated in a linear way. The considered learners have also a regularization parameter, which controls the tradeoff between model complexity and fit to the training data. In the experiments we did not find the level of regularization applied to have major effect on the relative quality of the cross-validation estimates, so we consider only the results for regularization parameter value 1. The used learning and cross-validation algorithms are from our RLScore software package, available at `http://www.tucs.fi/rlscore`. All the experiments are repeated 10000 times. We assess the significance of the difference between the deviation of the LPOCV estimate and the alternative estimates using the Wilcoxon signed-rank test, with $p = 0.05$, applying the Bonferroni correction for multiple hypothesis testing.

Figure 1 displays the results for non-signal data. When using the RLS- learner on low-dimensional data, we observe a substantial bias for the pooled estimators, with balanced LOOCV being the least biased of them. The averaging strategies work better, with LPOCV showing significantly less bias than all of the pooled strategies. These results are consistent with those reported in [8]. With RankRLS and low-dimensional data, the pessimistic bias of the pooled strategies is much smaller, but nonetheless significant differences compared to the less pessimistic LPOCV are observed. LPOCV and the other averaged strategies behave similarly. On high-dimensional data none of the estimates show clear bias.

Figure 2 displays the results for signal data. Again, with the RLS learner and low-dimensional data, a large pessimistic bias is present in the pooled estimates. LPOCV gives significantly less biased performance estimates. For RankRLS we

observe the same phenomenon, though the negative bias of the pooled strategies is much smaller than for RLS (similarly to the no-signal experiment). On high-dimensional data, most of the pessimistic bias seems to disappear from the pooled estimates. With RankRLS, LOOCV actually provides significantly more optimistic performance estimates than LPOCV, though the magnitudes of the differences in their mean deviations are very small. Of the averaged strategies, the bias of tenfold cross-validation is similar to that of LPOCV. However, averaged fivefold cross-validation is in most of the signal experiments much more pessimistically biased than LPOCV.

In all of the experiments, averaged tenfold and fivefold strategies have larger variance than the pooled strategies and LPOCV. The more imbalanced the relative class distributions, the higher the variance becomes. This effect is magnified for averaged tenfold and fivefold, as folds which do not have examples from both classes can not be considered when calculating the average AUC.

To conclude, LPOCV shows very little bias in both low- and high dimensional feature space, and has a very similar variance to that of the pooled strategies. Averaged tenfold cross-validation is also very competitive in terms of bias, but suffers from large variance, as does averaged fivefold cross-validation. Furthermore, for averaged fivefold large pessimistic bias appears in the signal experiment. This is probably due to the fact that one fifth of the training data is held out of the already very small training set in each round. LOOCV and balanced LOOCV worked well in many settings, but both suffered from a large negative bias on low-dimensional data and RLS learner.

4 Conclusion

In this work we have considered the merits and drawbacks of different conditional expected AUC cross-validation estimators, in the small sample setting. In terms of variance, the averaged fivefold and tenfold cross-validation proved to be inferior to the pooled strategies and LPOCV. On low dimensional data sets, large negative bias was observed in the pooled estimators showing that they can systematically fail in such a setting. However, with increased dimensionality this effect disappeared, suggesting that the pooled estimators can be very competitive when using high dimensional data. LPOCV seems to be overall the most robust method, as it is in all settings almost unbiased, and shows variance that is competitive with that of the pooled estimators.

Based on the simulation results we suggest the use of LPOCV for AUC-estimation due to its robustness. For RLS based learners calculating the LPOCV can be done efficiently, for other types of methods the computational cost can be high. Further study is needed to ascertain whether the large bias exhibited by the pooled estimators is a phenomenom that appears only when dealing with small dimensional data. If this is the case, the pooled CV strategies may also be considered suitable for AUC estimation for high dimensional data, which is a typical property of data produced by biomolecular studies.

Acknowledgments

This work has been supported by the Academy of Finland. W.W. was supported by a research visit grant from the Research Foundation Flanders.

References

1. Bradley, A.P.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recogn. **30**(7) (1997) 1145–1159
2. Waegeman, W., De Baets, B., Boullart, L.: ROC analysis in ordinal regression learning. Pattern Recogn. Lett. **29**(1) (2008) 1–9
3. Vanderlooy, S., Hüllermeier, E.: A critical analysis of variants of the AUC. Mach. Learn. **72**(3) (2008) 247–262
4. Baker, S., Kramer, B.: Identifying genes that contribute most to good classification in microarrays. BMC Bioinformatics **7**(1) (2006)
5. Gevaert, O., De Smet, F., Timmerman, D., Moreau, Y., De Moor, B.: Predicting the prognosis of breast cancer by integrating clinical and microarray data with bayesian networks. Bioinformatics **22**(14) (2006) 184–190
6. Swets, J.: Measuring the accuracy of diagnostic systems. Science **240**(4857) (1988) 1285–1293
7. Airola, A., Pyysalo, S., Björne, J., Pahikkala, T., Ginter, F., Salakoski, T.: All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. BMC Bioinformatics **9**(Suppl 11) (2008) S2
8. Parker, B.J., Gunter, S., Bedo, J.: Stratification bias in low signal microarray studies. BMC Bioinformatics **8**(326) (2007)
9. Cortes, C., Mohri, M., Rastogi, A.: An alternative ranking problem for search engines. In: Proceedings of WEA'07. (2007) 1–21
10. Agarwal, S., Graepel, T., Herbrich, R., Har-Peled, S., Roth, D.: Generalization bounds for the area under the ROC curve. J. Mach. Learn. Res. **6** (2005) 393–425
11. Braga-Neto, U.M., Dougherty, E.R.: Is cross-validation valid for small-sample microarray classification? Bioinformatics **20**(3) (2004) 374–380
12. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning, Second Edition. (2009)
13. Cortes, C., Mohri, M.: AUC optimization vs. error rate minimization. In Thrun, S., Saul, L., Schölkopf, B., eds.: Proceedings of NIPS'03. (2003)
14. Pahikkala, T., Airola, A., Boberg, J., Salakoski, T.: Exact and efficient leave-pair-out cross-validation for ranking RLS. In Honkela, T., Pöllä, M., Paukkeri, M.S., Simula, O., eds.: Proceedings of AKRR'08. (2008) 1–8
15. Rifkin, R., Yeo, G., Poggio, T.: Regularized least-squares classification. In Suykens, J., Horvath, G., Basu, S., Micchelli, C., Vandewalle, J., eds.: Advances in Learning Theory: Methods, Model and Applications. (2003) 131–154
16. Pahikkala, T., Tsivtsivadze, E., Airola, A., Boberg, J., Järvinen, J.: An efficient algorithm for learning to rank from preference graphs. Mach. Learn. **75**(1) (2009) 129–165
17. Pahikkala, T., Boberg, J., Salakoski, T.: Fast n-fold cross-validation for regularized least-squares. In Honkela, T., Raiko, T., Kortela, J., Valpola, H., eds.: Proceedings of SCAI'06. (2006) 83–90
18. Zhang, P., Peng, J.: SVM vs regularized least squares classification. In Kittler, J., Petrou, M., Nixon, M., eds.: Proceedings of ICPR'04. (2004) 176–179

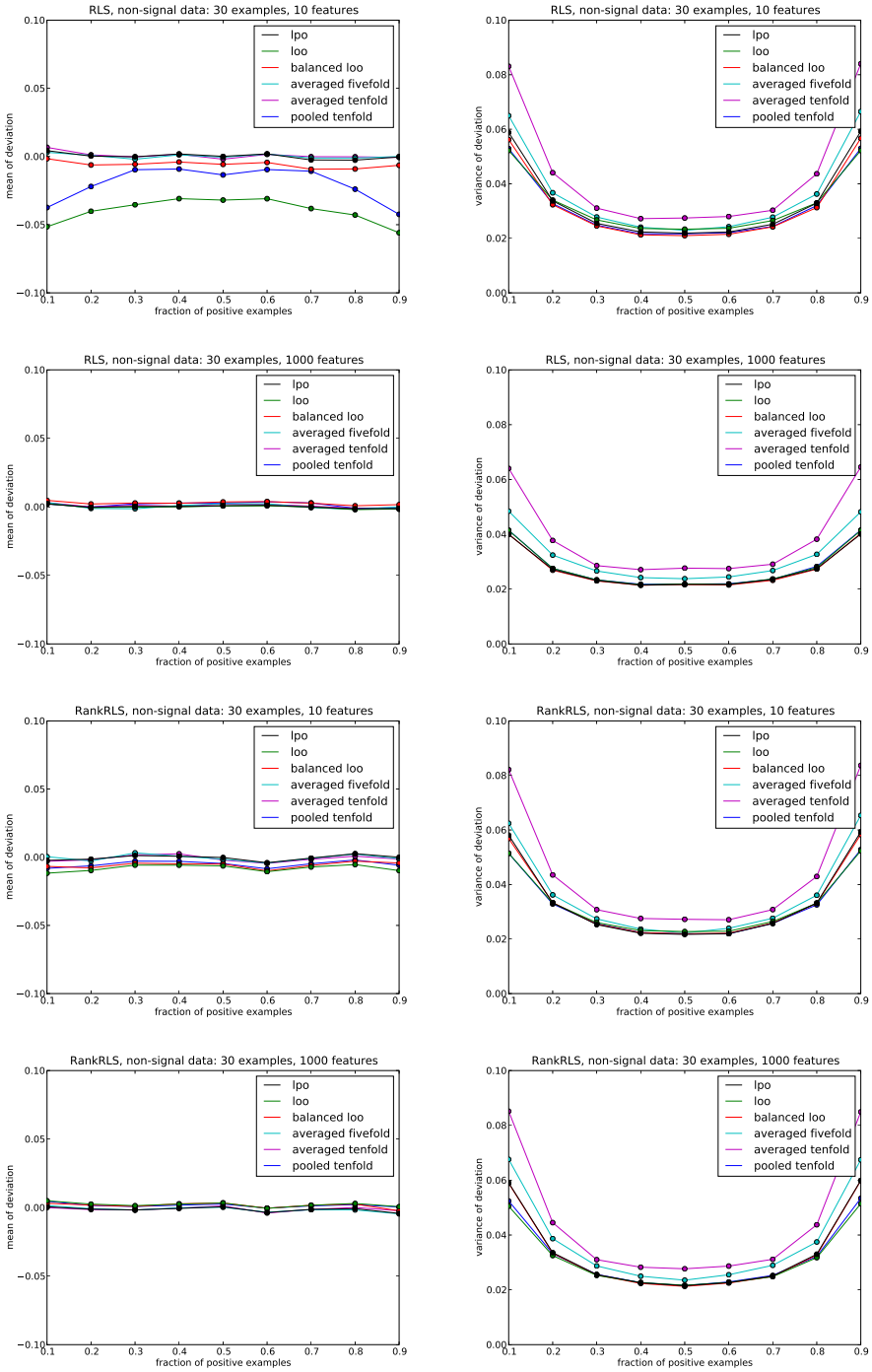

Fig. 1. Mean and variance of the deviation distribution for the non-signal data.

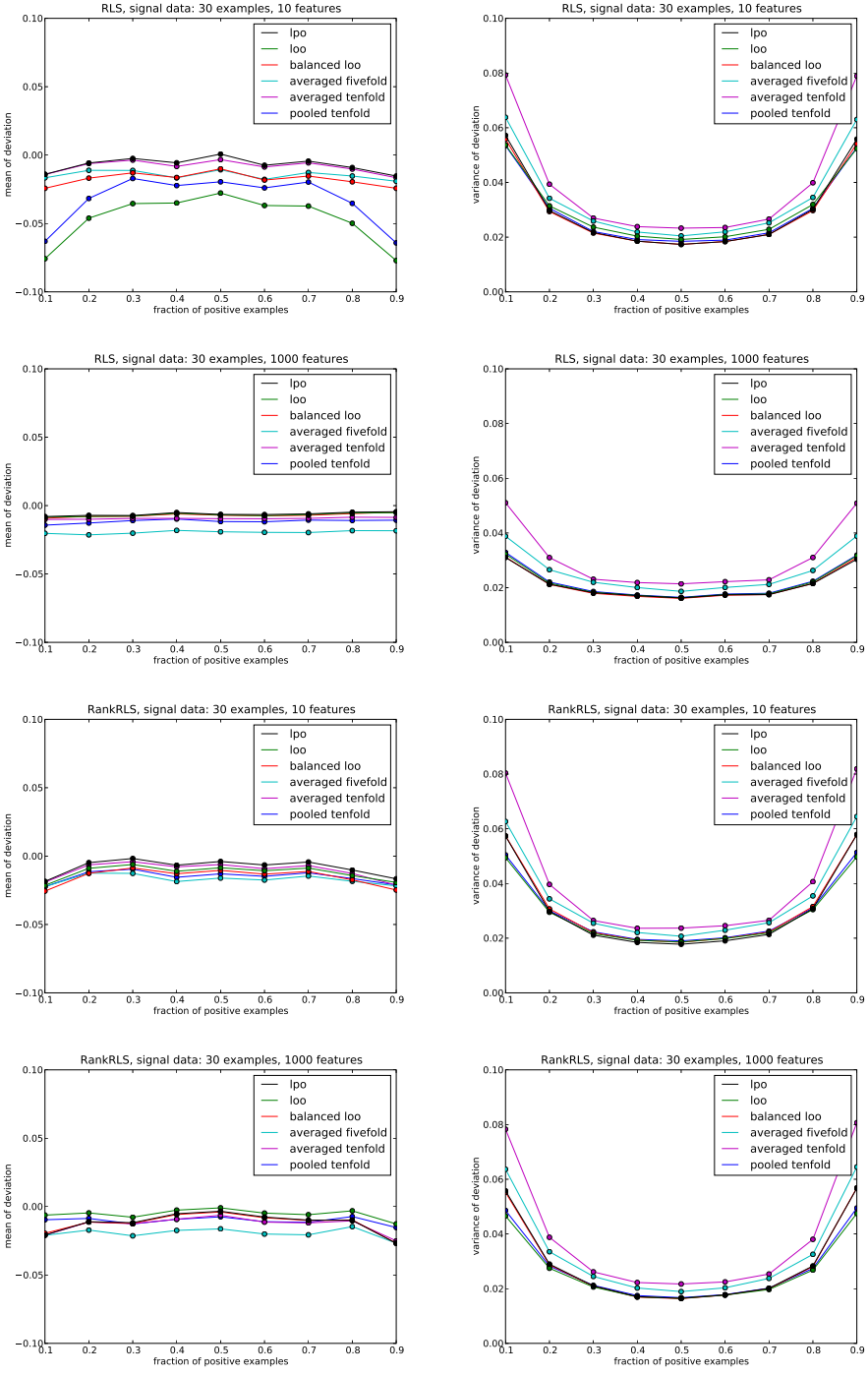

Fig. 2. Mean and variance of the deviation distribution for the signal data.