Validation of Prognostic Models based on High Dimensional Data

Kinds of Validation

- Analytical validation
  - Is the assay reproducible and robust?
  - Does it measure the analytes accurately?
- Clinical validation
  - Predictive accuracy
  - Sensitivity, specificity, ppv, npv, ROC
- Medical utility
  - Is the test actionable and improves outcome?

Samples Required for Validation

- Analytical
  - Samples collected and handled as for the intended use
  - No clinical annotation required
- Clinical
  - Performance measures for intended use
  - Usually retrospective
- Medical Utility
  - May require prospective study

Gene Expression–Based Prognostic Signatures in Lung Cancer: Ready for Clinical Use?

Jyothi Subramanian, Richard Simon

Manuscript received July 9, 2009; revised December 29, 2009; accepted January 15, 2010.

Correspondence to: Richard Simon, DSc, Biometric Research Branch, Department of Cancer Treatment and Diagnosis, National Cancer Institute, 9000 Rockville Pike, Bethesda, MD 20892-7434 (e-mail: rsimon@mail.nih.gov).

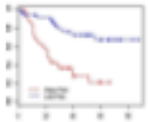
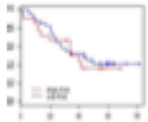
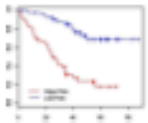
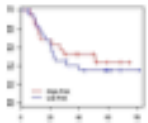
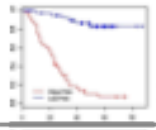
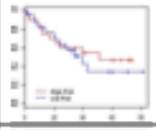
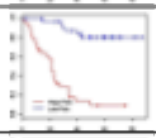
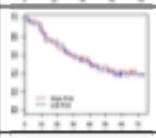
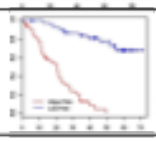
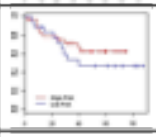
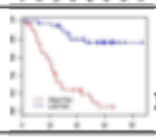
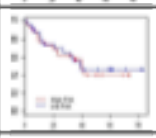
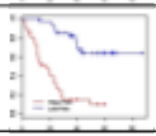
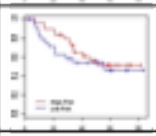
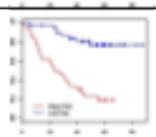
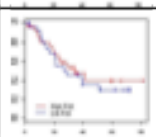
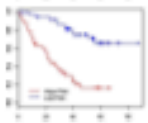
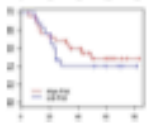
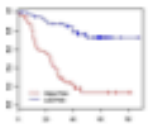
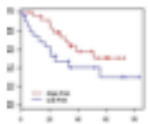
A substantial number of studies have reported the development of gene expression–based prognostic signatures for lung cancer. The ultimate aim of such studies should be the development of well-validated clinically useful prognostic signatures that improve therapeutic decision making beyond current practice standards. We critically reviewed published studies reporting the development of gene expression–based prognostic signatures for non–small cell lung cancer to assess the progress made toward this objective. Studies published between January 1, 2002, and February 28, 2009, were identified through a PubMed search. Following hand-screening of abstracts of the identified articles, 16 were selected as relevant. Those publications were evaluated in detail for appropriateness of the study design, statistical validation of the prognostic signature on independent datasets, presentation of results in an unbiased manner, and demonstration of medical utility for the new signature beyond that obtained using existing treatment guidelines. Based on this review, we found little evidence that any of the reported gene expression signatures are ready for clinical application. We also found serious problems in the design and analysis of many of the studies. We suggest a set of guidelines to aid the design, analysis, and evaluation of prognostic signature studies. These guidelines emphasize the importance of focused study planning to address specific medically important questions and the use of unbiased analysis methods to evaluate whether the resulting signatures provide evidence of medical utility beyond standard of care–based prognostic factors.

J Natl Cancer Inst 2010;102:464–474

Commonly Seen Problems

- Samples not collected for a clearly specified intended use

- Over-estimation of prediction accuracy
  - Develop prognostic classifier of survival on a dataset
  - Classify the cases of the same dataset as good prognosis or poor prognosis using the classifier
  - Calculate Kaplan Meier estimates of survival for each risk group

Simulation	Training		Validation	
1		p=7.0e-05		p=0.70
2		p=4.2e-07		p=0.54
3		p=2.4e-13		p=0.60
4		p=1.3e-10		p=0.89
5		p=1.8e-13		p=0.36
6		p=5.5e-11		p=0.81
7		p=3.2e-09		p=0.46
8		p=1.8e-07		p=0.61
9		p=1.1e-07		p=0.49
10		p=4.3e-09		p=0.09

Validation of a prognostic model

- Evaluating prediction accuracy of a completely "locked down" model on an independent dataset
- Splitting sample into training set and validation set
  - Split by center
- Cross-validation or bootstrap re-sampling to estimate prediction accuracy

- Split sampling and re-sampling provide internal estimates of prediction accuracy that may not reflect inter-laboratory assay variation and other sources of inter-center variation

Dobbin and Simon *BMC Medical Genomics* 2011, **4**:31
http://www.biomedcentral.com/1755-8794/4/31

BMC Medical Genomics

RESEARCH ARTICLE Open Access

Optimally splitting cases for training and testing high dimensional classifiers

Kevin K Dobbin[1*] and Richard M Simon[2]

Abstract

Background: We consider the problem of designing a study to develop a predictive classifier from high dimensional data. A common study design is to split the sample into a training set and an independent test set, where the former is used to develop the classifier and the latter to evaluate its performance. In this paper we address the question of what proportion of the samples should be devoted to the training set. How does this proportion impact the mean squared error (MSE) of the prediction accuracy estimate?

Results: We develop a non-parametric algorithm for determining an optimal splitting proportion that can be applied with a specific dataset and classifier algorithm. We also perform a broad simulation study for the purpose of better understanding the factors that determine the best split proportions and to evaluate commonly used splitting strategies (1/2 training or 2/3 training) under a wide variety of conditions. These methods are based on a decomposition of the MSE into three intuitive component parts.

Conclusions: By applying these approaches to a number of synthetic and real microarray datasets we show that for linear classifiers the optimal proportion depends on the overall number of samples available and the degree of differential expression between the classes. The optimal proportion was found to depend on the full dataset size (n) and classification accuracy - with higher accuracy and smaller *n* resulting in more assigned to the training set. The commonly used strategy of allocating 2/3rd of cases for training was close to optimal for reasonable sized datasets ($n \geq 100$) with strong signals (i.e. 85% or greater full dataset accuracy). In general, we recommend use of our nonparametric resampling approach for determing the optimal split. This approach can be applied to any dataset, using any predictor development method, to determine the best split.

Background

The split sample approach is a widely used study design in high dimensional settings. This design divides the collection into a training set and a test set as a means of estimating classification accuracy. A classifier is developed on the training set and applied to each sample in the test set. In practice, statistical prediction models have often been developed without separating the data used for model development from the data used for estimation of prediction accuracy [1]. When the number of candidate predictors (p) is larger than the number of cases as in microarray data, such separation is essential to avoid large bias in estimation of prediction accuracy [2]. This paper addresses the question of how to optimally split a sample into a training set and a test set for a high dimensional gene expression study, that is, how many samples to allocate to each group.

Two approaches to evaluating splits of the data are examined. The first approach is based on simulations designed to understand qualitatively the relationships among dataset characteristics and optimal split proportions. We use these results also to evaluate commonly used rules-of-thumb for allocation of the data to training and test sets. Our second approach involves development of a non-parametric method that does not rely on distributional assumptions and can be applied directly to any existing dataset without stipulating any parameter values. The nonparametric method can be used with any predictor development method (e.g., nearest neighbor, support vector machine).

This paper addresses the situation in which the accuracy of a predictor will be assessed by its

* Correspondence: dobbinke@uga.edu
[1]Department of Epidemiology and Biostatistics, College of Public Health, University of Georgia, Athens, GA, USA
Full list of author information is available at the end of the article

LOOCV for two class prediction of binary response

- Full dataset P={1,2,…,n}
- Omit case 1
  - V_1={1}; T_1={2,3,…,n}
  - Develop classifier using training set T_1
  - Classify cases in V_1 and count whether classification is correct or not
- Repeat for case 2,3,…
- Total number of mis-classified cases

Complete Cross-Validation

- Simulates the development of a model on one dataset and testing it on another

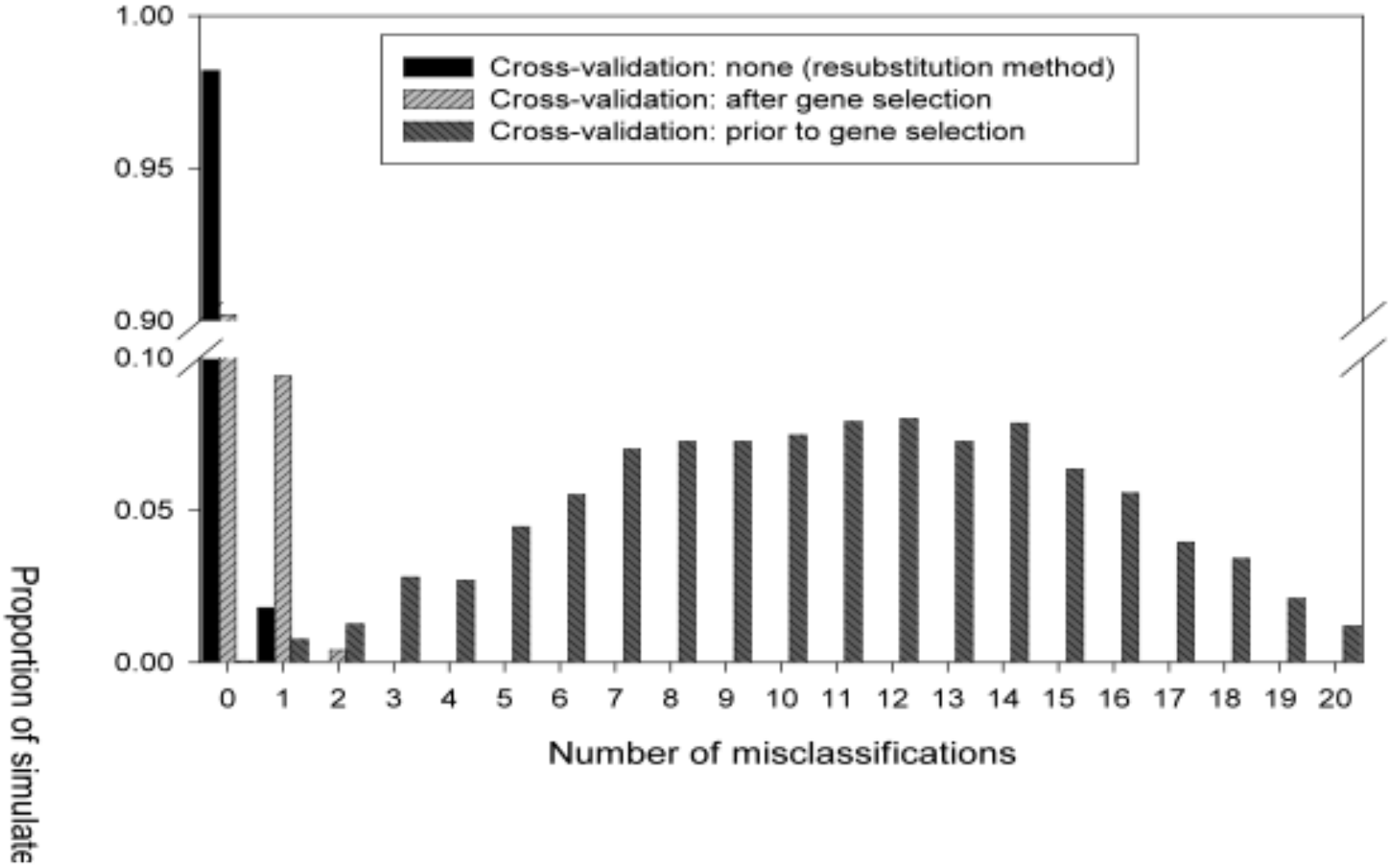
- All aspects of the model development process should be repeated from scratch for each loop of the cross-validation
  - Variable selection
  - Model fitting
  - Tuning parameter optimization
  - Setting threshold for classification

Prediction on Simulated Null Data

Simon et al. JNCI:95,14,2003

- Generation of gene expression profiles
  - 20 specimens, 6000 genes MVN(0,I_{6000})
  - 1st 10 specimens class 1, last 10 class 2
- Prediction method
  - Compound covariate classifier based on the 10 most differentially expressed genes

$$\sum_{\text{g selected}} t_g x_g$$


Cross-validation: none (resubstitution method)
Cross-validation: after gene selection
Cross-validation: prior to gene selection
1.00
0.95
0.90
0.10
0.05
0.00
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Number of misclassifications
Proportion of simulate

- Goodness of fit to the data used for building the model is not evidence of model accuracy
- Statistical significance of regression coefficients or Kaplan Meier curves is not evidence of model accuracy
- A hazard ratio is a measure of association, not prediction accuracy; large HR can mask poor prediction accuracy

BMC Bioinformatics

BioMed Central

Research article

Open Access

Bias in error estimation when using cross-validation for model selection

Sudhir Varma*† and Richard Simon†

Address: Biometric Research Branch, National Cancer Institute, Bethesda MD, USA

Email: Sudhir Varma* - varmas@mail.nih.gov; Richard Simon - rsimon@mail.nih.gov

* Corresponding author †Equal contributors

Published: 23 February 2006

BMC Bioinformatics 2006, **7**:91 doi:10.1186/1471-2105-7-91

Received: 28 April 2005

Accepted: 23 February 2006

This article is available from: http://www.biomedcentral.com/1471-2105/7/91

Abstract

Background: Cross-validation (CV) is an effective method for estimating the prediction error of a classifier. Some recent articles have proposed methods for optimizing classifiers by choosing classifier parameter values that minimize the CV error estimate. We have evaluated the validity of using the CV error estimate of the optimized classifier as an estimate of the true error expected on independent data.

Results: We used CV to optimize the classification parameters for two kinds of classifiers; Shrunken Centroids and Support Vector Machines (SVM). Random training datasets were created, with no difference in the distribution of the features between the two classes. Using these "null" datasets, we selected classifier parameter values that minimized the CV error estimate. 10-fold CV was used for Shrunken Centroids while Leave-One-Out-CV (LOOCV) was used for the SVM. Independent test data was created to estimate the true error. With "null" and "non null" (with differential expression between the classes) data, we also tested a nested CV procedure, where an inner CV loop is used to perform the tuning of the parameters while an outer CV is used to compute an estimate of the error.

The CV error estimate for the classifier with the optimal parameters was found to be a substantially biased estimate of the true error that the classifier would incur on independent data. Even though there is no real difference between the two classes for the "null" datasets, the CV error estimate for the Shrunken Centroid with the optimal parameters was less than 30% on 18.5% of simulated training data-sets. For SVM with optimal parameters the estimated error rate was less than 30% on 38% of "null" data-sets. Performance of the optimized classifiers on the independent test set was no better than chance.

The nested CV procedure reduces the bias considerably and gives an estimate of the error that is very close to that obtained on the independent testing set for both Shrunken Centroids and SVM classifiers for "null" and "non-null" data distributions.

Conclusion: We show that using CV to compute an error estimate for a classifier that has itself been tuned using CV gives a significantly biased estimate of the true error. Proper use of CV for estimating true error of a classifier developed using a well defined algorithm requires that all steps of the algorithm, including classifier parameter tuning, be repeated in each CV loop. A nested CV procedure provides an almost unbiased estimate of the true error.

- The cross-validated misclassification rate is an almost unbiased estimate of the generalization error for the classifier developed on the full sample when applied to future observations drawn from the same distribution
- C(x;T)=classifier developed on training set T
- C(x;D)=classifier developed on full dataset D
- $\mu(D)=E_{(x,y)}[I\{C(x;D)\neq y\}]$ generalization error rate
- Cross-validated misclassification rate $\approx E_D[\mu(D)]$

- Partition D into (T,V). Develop classifier C(;T)
- Evaluate C(;T) on V;

$$val_{split} = \frac{1}{|V|} \sum_{(x,y)\in V} I\{C(x;T) \neq y\}$$
$$= \hat{\mu}(T)$$

- For small samples, the cross-validated estimate is a less biased estimator of μ(D) than the split-sample estimate and has smaller MSE

BIOINFORMATICS ORIGINAL PAPER

Vol. 21 no. 15 2005, pages 3301–3307
doi:10.1093/bioinformatics/bti499

Data and text mining

Prediction error estimation: a comparison of resampling methods

Annette M. Molinaro[1,3,*], Richard Simon[2] and Ruth M. Pfeiffer[1]

[1]Biostatistics Branch, Division of Cancer Epidemiology and Genetics and [2]Biometric Research Branch, Division of Cancer Treatment and Diagnostics, NCI, NIH, Rockville, MD 20852 USA and [3]Department of Epidemiology and Public Health, Yale University School of Medicine, New Haven, CT 06520, USA

Received on April 6, 2005; revised on April 28, 2005; accepted on May 12, 2005
Advance Access publication May 19, 2005

ABSTRACT

Motivation: In genomic studies, thousands of features are collected on relatively few samples. One of the goals of these studies is to build classifiers to predict the outcome of future observations. There are three inherent steps to this process: feature selection, model selection and prediction assessment. With a focus on prediction assessment, we compare several methods for estimating the 'true' prediction error of a prediction model in the presence of feature selection.

Results: For small studies where features are selected from thousands of candidates, the resubstitution and simple split-sample estimates are seriously biased. In these small samples, leave-one-out cross-validation (LOOCV), 10-fold cross-validation (CV) and the .632+ bootstrap have the smallest bias for diagonal discriminant analysis, nearest neighbor and classification trees. LOOCV and 10-fold CV have the smallest bias for linear discriminant analysis. Additionally, LOOCV, 5- and 10-fold CV, and the .632+ bootstrap have the lowest mean square error. The .632+ bootstrap is quite biased in small sample sizes with strong signal-to-noise ratios. Differences in performance among resampling methods are reduced as the number of specimens available increase.

Contact: annette.molinaro@yale.edu

Supplementary Information: A complete compilation of results and R code for simulations and analyses are available in Molinaro *et al.* (2005) (http://linus.nci.nih.gov/brb/TechReport.htm).

Including too many noisy variables reduces accuracy of the prediction and may lead to over-fitting of data, resulting in promising but often non-reproducible results (Ransohoff, 2004).

Another difficulty is model selection with numerous classification models available. An important step in reporting results is assessing the chosen model's error rate, or generalizability. In the absence of independent validation data, a common approach to estimating predictive accuracy is based on some form of resampling the original data, e.g. cross-validation. These techniques divide the data into a learning set and a test set, and range in complexity from the popular learning-test split to v-fold cross-validation, Monte-Carlo v-fold cross-validation and bootstrap resampling. Few comparisons of standard resampling methods have been performed to date, and all of them exhibit limitations that make their conclusions inapplicable to most genomic settings. Early comparisons of resampling techniques in the literature are focussed on model selection as opposed to prediction error estimation (Breiman and Spector, 1992; Burman, 1989). In two recent assessments of resampling techniques for error estimation (Braga-Neto and Dougherty, 2004; Efron, 2004), feature selection was not included as part of the resampling procedures, causing the conclusions to be inappropriate for the high-dimensional setting.

We have performed an extensive comparison of resampling methods to estimate prediction error using simulated (large signal-to-noise ratio), microarray (intermediate signal to noise ratio) and proteomic data (low signal-to-noise ratio), encompassing increasing sample

Resampling with and without replacement To understand the ramification of resampling with replacement as it pertains to the bootstrap estimates, we compared the leave-one-out bootstrap estimate (Section 2.1.5) to the 3-fold MCCV. The 3-fold MCCV randomly selects .666n for the learning set and .333n for the test set. This is repeated numerous times and the estimates averaged. Therefore the 3-fold MCCV is equivalent to the leave-one-out bootstrap, except it employs resampling without replacement. Table 5 displays the simulation study results for the two estimates using 50 iterations for both. Interestingly, the bias and MSE for the leave-one-out bootstrap are roughly double that of 3-fold MCCV. The only two distinct differences between the two methods are the replicate copies in the learning set, inherent in the bootstrap estimate, and the fact that **on average** .632n unique observations are in the learning sample for the leave-one-out bootstrap, whereas there are always .666n in the learning sample for the 3-fold MCCV. Both these factors may contribute to the increase in bias and MSE.

4 DISCUSSION

Estimation of prediction error when confronted with a multitude of covariates and small sample sizes is a relatively new problem. Feature selection, sample size and signal-to-noise ratio are important influences on the relative performance of resampling methods. We have evaluated resampling methods for use in high dimensional classification problems using a range of sample sizes, algorithms and signals. Some general conclusions may be summarized as follows:

(1) **With small sample sizes, the split sample method and 2-fold CV perform very poorly**. This poor performance is primarily due to a large positive bias resulting from the use of a reduced training set size, which severely impairs its ability to effectively select features and fit a model. The large bias contributes to a large MSE.

(2) **LOOCV generally performs very well with regard to MSE and bias**. The only exception is when an unstable classifier (e.g. CART) is used in the presence of a weak signal. In this setting, the larger MSE is attributed to LOOCV's increased variance.

(3) **10-fold CV prediction error estimates approximate those of LOOCV in almost all settings**. For computationally burdensome analyses, 10-fold CV may be preferable to LOOCV. Additionally, in the simulated data, repeated resamplings (the average of 10 repeats) reduce the MSE, bias, and variance of 10-fold CV.

(4) **The .632+ prediction error estimate performs best with moderate to weak signal-to-noise ratios**. Previous studies have found the bootstrap variants superior to LOOCV and v-fold CV; however, these studies did not include feature selection. As seen in Table 1, honest resampling in small samples with strong signal suggest that LOOCV and 10-fold CV are in fact better than the .632+ bootstrap. This discrepancy fades when feature selection is discarded (Table 4) and when the signal decreases, as seen in the lymphoma and ovarian datasets (Tables 2 and 3). Additional glimpses into the bootstrap estimate (Table 5) indicate that the sampling with replacement increases the MSE and bias substantially over 3-fold MCCV (i.e. resampling without replacement).

(5) MCCV does not decrease the MSE or bias enough to warrant its use over v-fold CV.

(6) As the sample size grows, the differences among the resampling methods decrease. Additionally, as the signal decreases from strong in the simulated data to rather weak in the ovarian data the discrepancies between the methods diminish.

In future work we will compare the resampling methods for continuous outcomes and continue to explore the behavior of the bootstrap estimates. Also, the effect of feature selection method may play an important role in prediction and needs further investigation.

ACKNOWLEDGEMENTS

A.M.M. was supported by the Cancer Prevention Fellowship Program, DCP/NCI/NIH. The authors thank Mark J. van der Laan for fruitful discussions.

Conflict of Interest: none declared.

Table 2. Lymphoma study results

Resampling method	$n = 40$			$n = 80$			$n = 120$		
	SD	Bias	MSE	SD	Bias	MSE	SD	Bias	MSE
2-fold CV	0.085	0.038	0.01	0.043	0.002	0.004	0.031	0.0	0.003
5-fold CV	0.07	0.004	0.007	0.045	−0.008	0.005	0.032	−0.006	0.003
10-fold CV	0.063	−0.007	0.006	0.036	−0.009	0.003	0.031	−0.006	0.003
LOOCV	0.072	−0.019	0.008	0.04	−0.013	0.004	0.033	−0.004	0.003
Split 1/3	0.119	0.001	0.017	0.071	0.0	0.007	0.059	−0.004	0.005
Split 1/2	0.117	0.037	0.018	0.058	0.001	0.005	0.046	−0.001	0.004
.632+	0.049	−0.006	0.004	0.025	−0.02	0.003	0.018	−0.015	0.002

Comparison of resampling method's MSE, bias and SD. Results shown are for the DDA algorithm using the top 10 genes as ranked by t-tests.

Table 1. Prediction error estimates

Estimator	p	Algorithm	Estimate	SD	Bias	MSE
$\tilde{\theta}_n$	0.87	LDA	0.078	0.093		
		DDA	0.160	0.086		
		NN	0.042	0.084		
		CART	0.121	0.133		
v-fold CV	0.5	LDA	0.357	0.126	0.279	0.097
		DDA	0.342	0.106	0.182	0.052
		NN	0.277	0.135	0.235	0.077
		CART	0.430	0.121	0.309	0.134
	0.2	LDA	0.161	0.127	0.083	0.017
		DDA	0.208	0.086	0.048	0.012
		NN	0.108	0.102	0.066	0.011
		CART	0.284	0.117	0.163	0.055
	0.1	LDA	0.118	0.120	0.040	**0.008**
		DDA	0.177	0.087	0.017	**0.007**
		NN	0.078	0.102	0.036	**0.005**
		CART	0.189	0.104	0.068	0.024
LOOCV	0.025	LDA	0.092	0.115	**0.014**	**0.008**
		DDA	0.164	0.096	**0.004**	**0.007**
		NN	0.058	0.103	**0.016**	**0.005**
		CART	0.146	0.125	**0.025**	**0.018**
Split	0.333	LDA	0.205	0.184	0.127	0.053
		DDA	0.243	0.138	0.083	0.034
		NN	0.145	0.169	0.103	0.044
		CART	0.371	0.174	0.25	0.121
	0.5	LDA	0.348	0.185	0.270	0.113
		DDA	0.344	0.139	0.184	0.062
		NN	0.265	0.177	0.223	0.086
		CART	0.438	0.155	0.317	0.147
.632+ 50 repetitions	$\approx$.368	LDA	0.274	**0.084**	0.196	0.047
		DDA	0.286	**0.074**	0.126	0.028
		NN	0.200	**0.070**	0.158	0.032
		CART	0.387	**0.080**	0.266	0.100

The estimate $\hat{\theta}_n$ (column 4) and SD (column 5) based on learning sample of size 40. The estimate $\tilde{\theta}_n$ (rows 1–4) and SD based on the remaining 260 observations. Bias (column 6) and MSE (column 7) reported for each resampling technique (column 1) and algorithm (column 3). The ten features with largest t-statistics used in algorithms. Minimums in bold.

Cross-Validated Performance Measures for Penalized Logistic Model Prediction

- Partition full dataset D into $D_1, D_2, \ldots, D_K$
- First training set $T_1 = D - D_1$
- Fit penalized logistic model to T_1 yielding model with regression coefficient vector $\beta^{(1)}$
- For cases $j \varepsilon D_1$ compute predictive score $s_j = \beta^{(1)} x_j$ and response probability $p_j = \exp(s_j)/(1+\exp(s_j))$
- Repeat for $T_2 = D - D_2, \ldots, T_K = D - D_K$

$$\text{sensitivity(c)} = \frac{\sum_{j=1}^{n} I(y_j = 1) I\left(p_j \geq c\right)}{\sum_{j=1}^{n} I(y_j = 1)}$$

$$\text{specificity(c)} = \frac{\sum_{j=1}^{n} I(y_j = 0) I\left(p_j < c\right)}{\sum_{j=1}^{n} I(y_j = 0)}$$

$$\text{ppv(c)} = \frac{\sum_{j=1}^{n} I(y_j = 1) I\left(p_j \geq c\right)}{\sum_{j=1}^{n} I(p_j \geq c)}$$

$$\text{npv(c)} = \frac{\sum_{j=1}^{n} I(y_j = 0) I\left(p_j < c\right)}{\sum_{j=1}^{n} I(p_j < c)}$$

BRIEFINGS IN BIOINFORMATICS. VOL 12. NO 3. 203–214
Advance Access published on 15 February 2011
doi:10.1093/bib/bbr001

Using cross-validation to evaluate predictive accuracy of survival risk classifiers based on high-dimensional data

Richard M. Simon, Jyothi Subramanian, Ming-Chung Li and Supriya Menezes

Submitted: 5th October 2010; Received (in revised form): 7th January 2011

Abstract

Developments in whole genome biotechnology have stimulated statistical focus on prediction methods. We review here methodology for classifying patients into survival risk groups and for using cross-validation to evaluate such classifications. Measures of discrimination for survival risk models include separation of survival curves, time-dependent ROC curves and Harrell's concordance index. For high-dimensional data applications, however, computing these measures as re-substitution statistics on the same data used for model development results in highly biased estimates. Most developments in methodology for survival risk modeling with high-dimensional data have utilized separate test data sets for model evaluation. Cross-validation has sometimes been used for optimization of tuning parameters. In many applications, however, the data available are too limited for effective division into training and test sets and consequently authors have often either reported re-substitution statistics or analyzed their data using binary classification methods in order to utilize familiar cross-validation. In this article we have tried to indicate how to utilize cross-validation for the evaluation of survival risk models; specifically how to compute cross-validated estimates of survival distributions for predicted risk groups and how to compute cross-validated time-dependent ROC curves. We have also discussed evaluation of the statistical significance of a survival risk model and evaluation of whether high-dimensional genomic data adds predictive accuracy to a model based on standard covariates alone.

Cross-Validated Kaplan-Meier Curves for Survival Risk Group PH Model

- Partition dataset D into $D_1, D_2, \ldots, D_K$
- First training set $T_1 = D - D_1$
- Develop survival risk classifier (e.g. penalized PH) using only T_1 yielding regression coefficient vector $\beta^{(1)}$
- Compute prognostic scores $\beta^{(1)}x$ for cases in D_1 and for cases in T_1 . For case j in D_1 if $\beta^{(1)}x_j < \text{median}\{\beta^{(1)}x_i , i \in T_1\}$ then predict case j is low risk; otherwise high risk.

- Repeat for T_2=D-D_2 etc.

- After K loops, all cases have been classified as low or high risk using a classifier developed on a training set that they were not part of.

- Compute Kaplan Meier curves for the two risk groups.

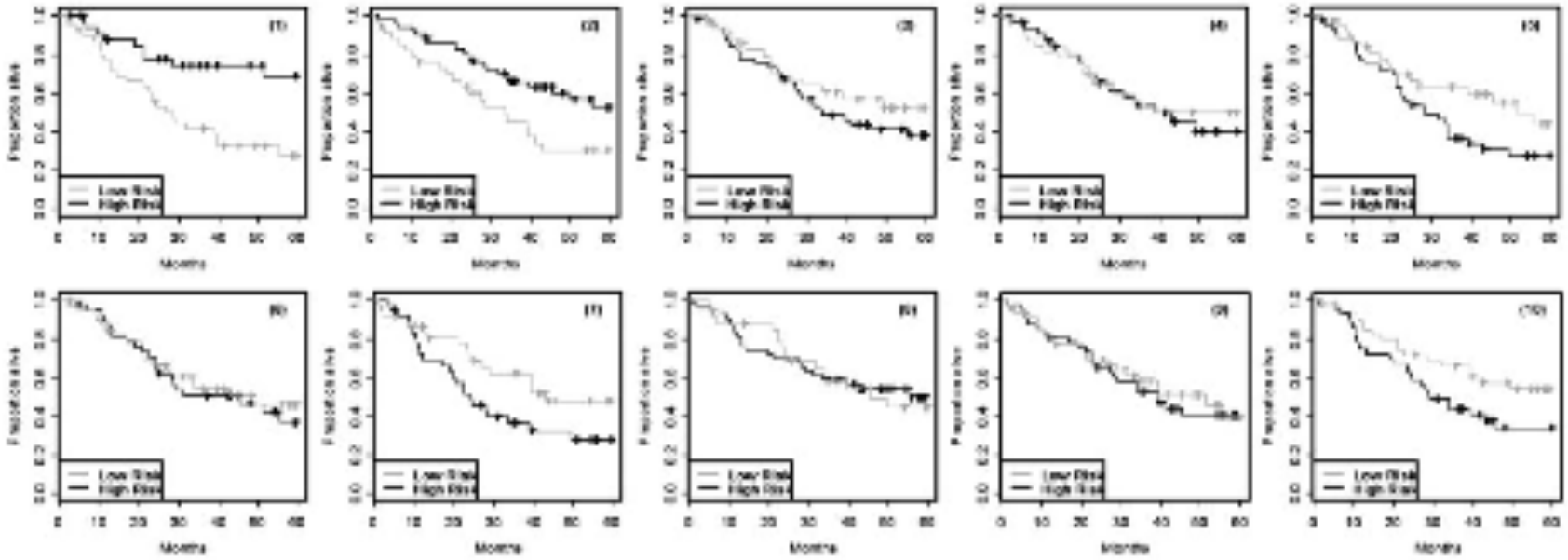


Figure 3: Cross-validated Kaplan–Meier survival estimates for the training sets shown in Figure 1.

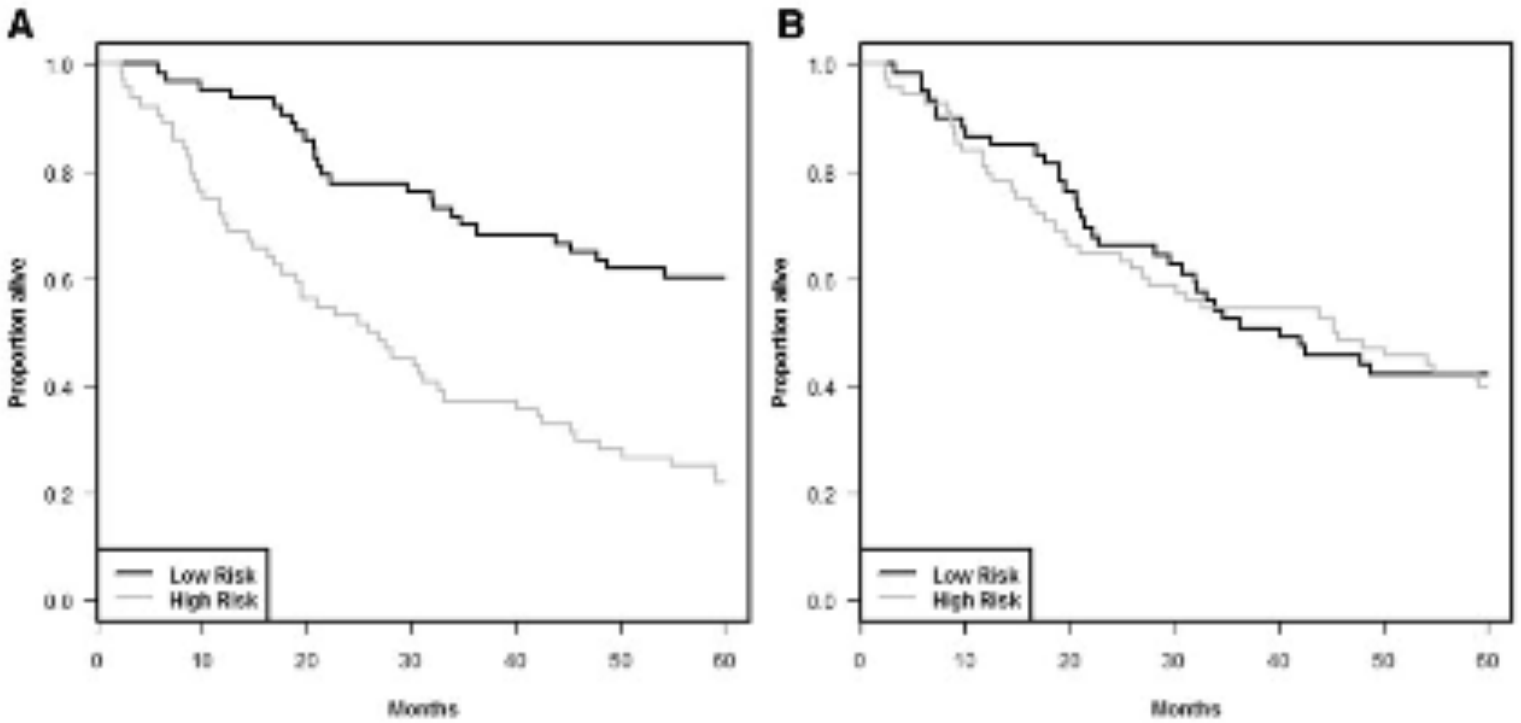


Figure 4: Kaplan–Meier survival curves for the data from Shedden *et al.* [18]. (**A**) Re-substitution estimates and (**B**) cross-validated estimates.

- To evaluate statistical significance of the spread of the cross-validated survival curves, the log-rank test cannot be used because the patient group indicators are random variables
- The null distribution of the log-rank statistic can be approximated, however, by permuting the survival times and repeating the entire procedure

- For each permutation, the cross-validation is repeated and new cross-validated KM curves produced. The log-rank statistic is calculated. This is repeated 1000 times to approximate the null distribution of the log-rank statistic.

Cross-Validated Time-Dependent ROC Curve

- Instead of computing cross-validated risk indicators, compute cross-validated quantile marker levels M_1, …, M_n

Cross-Validated Kaplan-Meier Curves for Survival Risk Group PH Model

- Partition dataset D into $D_1, D_2, \ldots, D_K$
- First training set $T_1 = D - D_1$
- Develop survival risk classifier (e.g. penalized PH) using only T_1 yielding regression coefficient vector $\beta^{(1)}$
- Compute prognostic scores $\beta^{(1)}x$ for cases in D_1 and for cases in T_1 . For case j in D_1
  M_j = rank of $\beta^{(1)}x_j$ in $\{\beta^{(1)}x_i$, all i$\}$

Time Dependent ROC Curve for Survival Data (Heagerty et. Al)

- Sensitivity(c) vs 1-Specificity(c)
- Marker M
- Cut-point c
- Sensitivity = $Pr[M \geq c \mid D=1]$
- Specificity = $Pr[M<c \mid D=0]$

Time Dependent ROC Curve for Survival Data (Heagerty et. Al)

- Sensitivity(c) vs 1-Specificity(c)
- Marker M
- Cut-point c
- Landmark time t*
- Sensitivity = $Pr[M \geq c \mid T \geq t^*]$
- Specificity = $Pr[M < c \mid T < t^*]$

Time Dependent ROC Curve

t^* = landmark time of interest

sensitivity(c)=Pr[βx>c|t ≤ t*]

$$= \frac{\Pr[t<t^*|\beta x>c]\Pr[\beta x>c]}{\Pr[t \le t^*]}$$

Estimate 1st term in numerator using KM estimator for cases with $\hat{\beta}_{-1} x_i > c$. Estimate 2nd term as proportion of n cases with $\hat{\beta}_{-1} x_i > c$. Estimate denominator using KM estimator for all n cases.

specificity(c)=Pr[βx ≤ c|t>t*]

$$= \frac{\Pr[t>t^*|\beta x \le c]\Pr[\beta x \le c]}{\Pr[t>t^*]}$$

Time-dependent ROC is sensitivity(c) vs 1-specificity(c)

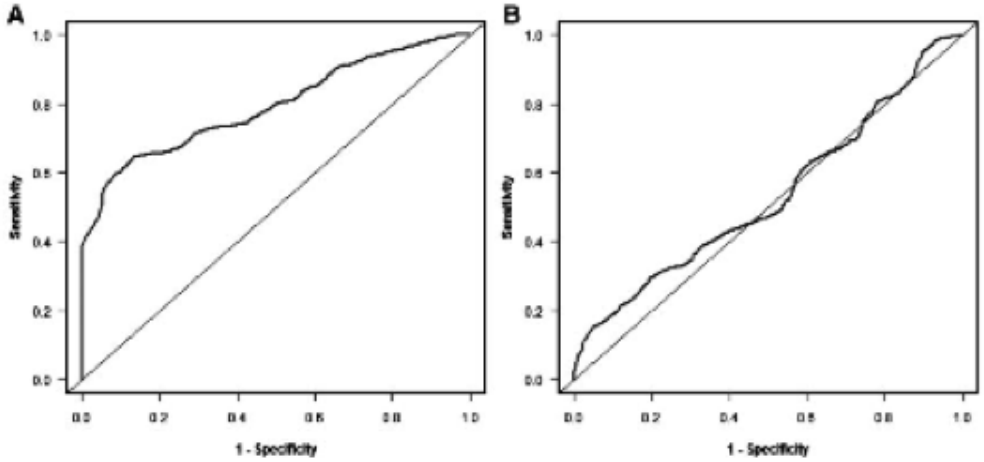


Figure 5: Time dependent ROC curves for the data from Shedden *et al.* [18]. (**A**) Re-substitution estimates and (**B**) cross-validated estimates. The resubstitution area under the curve (AUC) is 0.79 and the cross-validated AUC is 0.53.

Research Article

Statistics in Medicine

Received 20 January 2010, Accepted 13 September 2010 Published online 1 December 2010 in Wiley Online Library

(wileyonlinelibrary.com) DOI: 10.1002/sim.4106

An evaluation of resampling methods for assessment of survival risk prediction in high-dimensional settings[‡,§]

Jyothi Subramanian and Richard Simon[*†]

Resampling techniques are often used to provide an initial assessment of accuracy for prognostic prediction models developed using high-dimensional genomic data with binary outcomes. Risk prediction is most important, however, in medical applications and frequently the outcome measure is a right-censored time-to-event variable such as survival. Although several methods have been developed for survival risk prediction with high-dimensional genomic data, there has been little evaluation of the use of resampling techniques for the assessment of such models. Using real and simulated datasets, we compared several resampling techniques for their ability to estimate the accuracy of risk prediction models. Our study showed that accuracy estimates for popular resampling methods, such as sample splitting and leave-one-out cross validation (Loo CV), have a higher mean square error than for other methods. Moreover, the large variability of the split-sample and Loo CV may make the point estimates of accuracy obtained using these methods unreliable and hence should be interpreted carefully. A k-fold cross-validation with $k=5$ or 10 was seen to provide a good balance between bias and variability for a wide range of data settings and should be more widely adopted in practice. Published in 2010 by John Wiley & Sons, Ltd.

Probabilistic Binary Prediction

$$\log \frac{p}{1-p} = \beta' x$$

Penalized likelihood estimates $\hat{\beta}$

$$\hat{p} = \exp(\hat{\beta}' x) / [1 + \exp(\hat{\beta}' x)]$$

Biostatistics Advance Access published November 17, 2010

Biostatistics (2010), **0**, 0, *pp*. 1–14
doi:10.1093/biostatistics/kxq069

Probabilistic classifiers with high-dimensional data

KYUNG IN KIM, RICHARD SIMON*

Biometric Research Branch, National Cancer Institute,
9000 Rockville Pike, MSC 7434, Bethesda, MD 20892-7434, USA
rsimon@mail.nih.gov

SUMMARY

For medical classification problems, it is often desirable to have a probability associated with each class. Probabilistic classifiers have received relatively little attention for small n large p classification problems despite of their importance in medical decision making. In this paper, we introduce 2 criteria for assessment of probabilistic classifiers: well-calibratedness and refinement and develop corresponding evaluation measures. We evaluated several published high-dimensional probabilistic classifiers and developed 2 extensions of the Bayesian compound covariate classifier. Based on simulation studies and analysis of gene expression microarray data, we found that proper probabilistic classification is more difficult than deterministic classification. It is important to ensure that a probabilistic classifier is well calibrated or at least not "anticonservative" using the methods developed here. We provide this evaluation for several probabilistic classifiers and also evaluate their refinement as a function of sample size under weak and strong signal conditions. We also present a cross-validation method for evaluating the calibration and refinement of any probabilistic classifier on any data set.

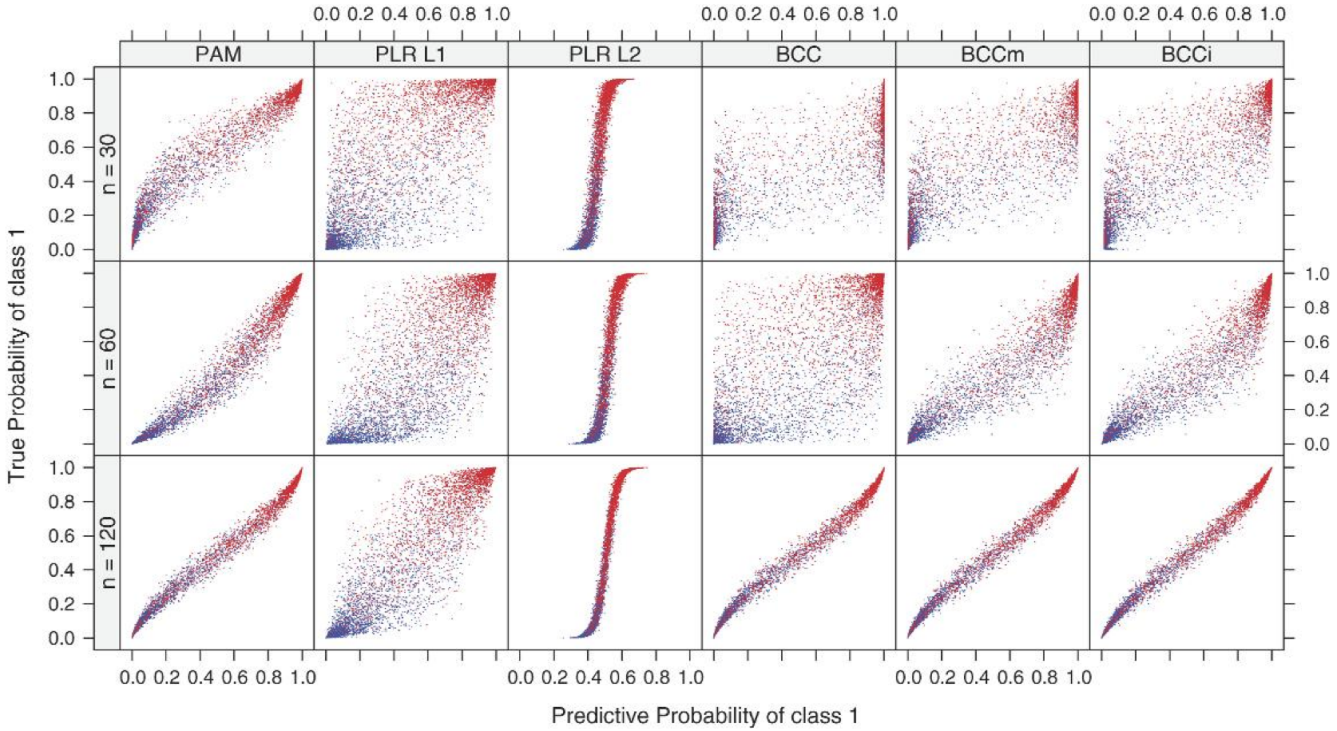


Fig. 1. A scatter plot for "Structure.1" condition. Rows represent 3 different training sample size ($n = 30, 60, 120$) and columns represent 6 probabilistic classifiers used to compute predictive probabilities. In each plot, x-axis represents predictive probability of class 1 and y-axis represents true probability of class 1. Each scatter consists of 5000 dots, where red ones are of class 1 and blue ones are of class 0. Equal prior class probability was assumed. Expression vectors are 1000-dimensional and normally distributed. Nonzero common pairwise correlation 0.25 was given only to informative 50 genes.

Well calibrated Probabilistic Classifier

$$\hat{p}^{-1}(u) = \left\{ x \ni \hat{p}(x) \in (u-\varepsilon, u+\varepsilon) \right\}, u \in (0,1)$$

$$\int p(x) dF(x \mid x \in \hat{p}^{-1}(u)) \simeq u$$

Evaluation of calibration

- Compute LOOCV estimate $\hat{p}_{-i}(x_i)$ for i=1,...,n
- Plot $\hat{p}_{-i}(x_i)$ versus the proportion of responders in cases j for which $\hat{p}_{-j}(x_j) \in (\hat{p}_{-i}(x_i) - \varepsilon, \hat{p}_{-i}(x_i) + \varepsilon)$

Evaluation of calibration

- Perform linear logistic regression of

$$(y_i, \hat{p}_{-i}(x_i))$$

- Slope 1 and intercept 0 indicates perfect calibration