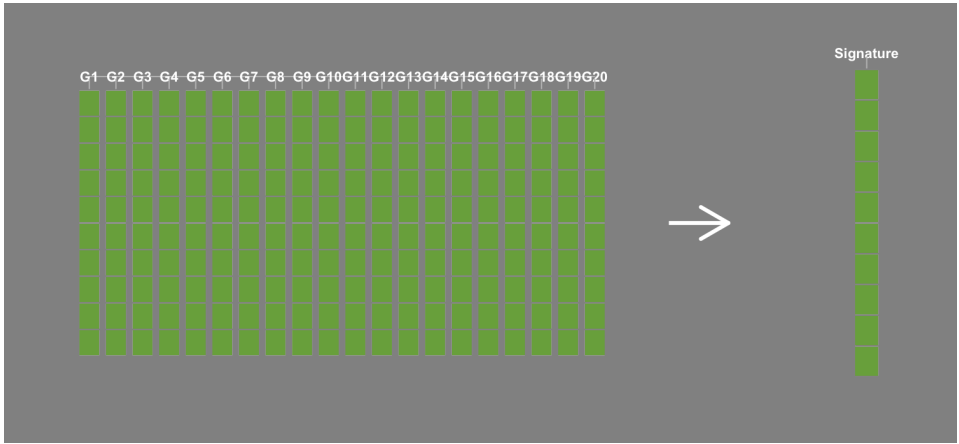
Predictive Modeling

Lecture III

Michael C Sachs

Predictive signatures

Basic principles



A signature

*A **biomarker signature** is a transformation of multiple individual features to a one-dimensional space.*

- A signature that *reliably predicts* an outcome *may* be useful for treatment selection or prognosis (actions?).
- A signature that *discriminates* between groups that would be treated differently may be clinically useful (utility?).
- A signature may be continuous, binary, or take multiple discrete values.

The performance of the signature is evaluated with the same measures we just discussed, but ...

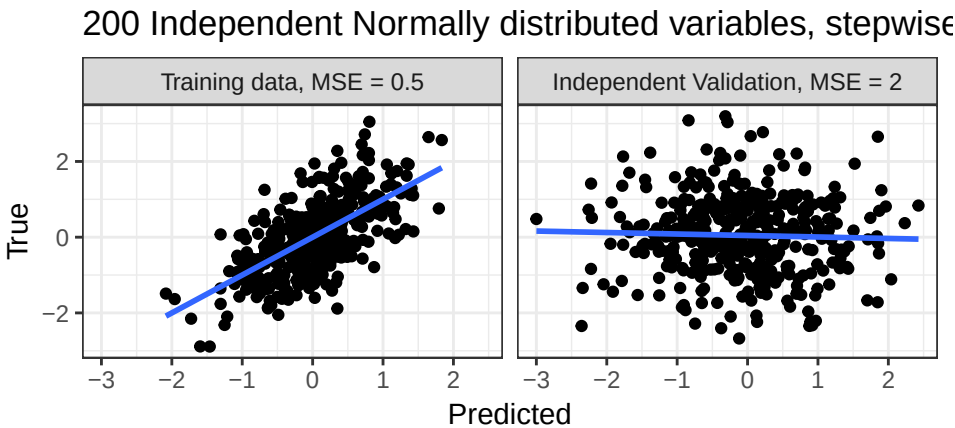
The development of a signature must be cleanly separated from its evaluation

Examples

1. We measure a biomarker, plot the ROC curve, and choose the best cutoff (upper left corner) to form dichotomous predictions
2. Measure a biomarker, estimate its regression model with outcome, and form continuous predictions
3. Measure 500,000 gene expression values, choose the 10 most significantly associated, and form a multivariable regression model with those.
4. Give a 10 item questionnaire, form predictions of 4 categories based on flow of responses to determine tax bracket.

All of these are estimate based on some training data. It is dishonest to evaluate the performance of the predictions using that same training data.

In-sample versus out-of-sample



Phases of biomarker signature development

- Development Phase Goals:
  - Estimate the signature based on some *training data*
    - Based on association with outcome
    - ... interaction with treatment
    - ... "Natural" clusters
    - ... "Expert" knowledge
  - Provide a **valid** estimate of performance of the signature
    - Depends on the true signal in the data
    - and the manner in which the signature is estimated
  - Provide a specification for others to use (optional)

Prediction error

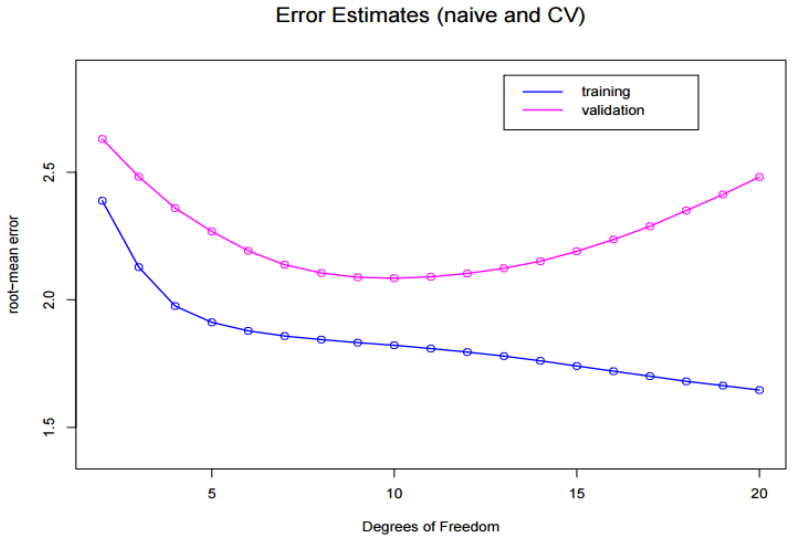
- More features means that the potential model complexity is high
- More parameters in the model $\rightarrow$ greater complexity
- Greater complexity $\rightarrow$ closer fit to the training sample

  Key point: *Random features in the training sample may not be reflected in the population*

The only honest estimate of prediction performance is based on data NOT used to develop the signature

- In multivariable linear regression, the in-sample R squared value (and hence MSE) will decrease with each additional variable in the model.
- This does not imply that the model has good out-of-sample prediction

Bias variance trade-off



Methods to get honest performance estimates

Let S denote the development dataset,

- Subscripts on S will denote different partitions of it
- The model building process generates a particular signature f
- ϕ_f denotes the performance evaluation function (MSE, AUC, etc.)

We are interested in estimating $E[\phi_f(S)]$, the expected error of f under the process that generated our sample.

The in-sample empirical estimate: $\hat{E}[\phi_f(S)] = \frac{1}{n}\sum_{i=1}^{n}\phi_f(s_i)$.

However, if the analyst **interacts with** S in the definition of ϕ_f, the estimate will be biased (overfit).

A horrifying example

Let's say I'm conducting a study to develop a genomic classifier for predicting lung cancer progression.

1. I tried out a clustering method and locked down the model
2. Testing it out the validation data, it didn't work as well as expected
3. For individual subjects where the prediction was poor, I manually changed the value of the outcome so that it looks better.

Obviously not OK. Still, any type of interaction with the validation data will likely bias the results

Split sample validation

- Randomly partition S into S_t and S_h with sample sizes n_t and n_h
  - Randomly split the data into two subsets (training and test)
- Hide S_h from yourself
- Generate an f_t using S_t only
  - Develop the model using the training set
- Estimate the error using the model on the test set

Cross validation

- Randomly partition S into V disjoint subsets, each of size k.
- Take the first partition and set aside
- Fit the model using the remaining data $n - k$
  - Estimate error using the k samples set aside
- Repeat for all V partitions
- Average the error estimates

"Leave-k-out" cross-validation, or V-fold cross-validation

Bootstrap

- Randomly sample b from S, with replacement and set them aside
- Derive the model using S_{-b}
- Estimate the error using samples not selected S_b
- Repeat and average

Summary

Method	Partitions	Sampling
Split-table	1	Without replacement
Cross-validation	Many	Without replacement
Bootstrap	Many	With replacement

Some details

- Ratio of training to test data
  - Depends on true signal in data (Dobbin & Simon, 2011)
  - Standard choices are 1:1 or 2:1 (training to test)
- Number of cross validation partitions ("leave-k-out")
  - Variation in signature estimation over folds
  - Instability of performance estimates
- Bootstrap
  - Sampling with replacement (put the drawn numbers back in the hat)
  - Many variations: Google "0.632+ bootstrap"

The Common Task Framework

- Take a subset of the sample, and hide it, ignore it, pretend the outcome doesn't exist. Give it to an "honest broker"
- Use rest of the sample to develop the model, multiple investigators can do it, trying different things
- Groups provide specifications of the models to the honest broker
- Broker forms predictions and evaluates performance

Netflix prize, Kaggle, DREAM challenges

Common error 1

Incomplete validation

- Use the full dataset to filter features down to a reasonable number using strength of the association with outcome
- Proceed to split-sample/cross/bootstrap validate the method for combining the selected features

BIASED!

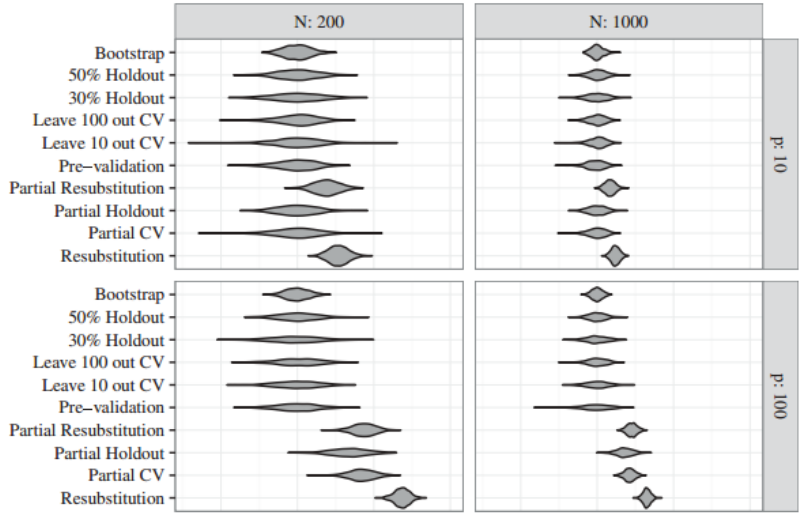
Common error 2

Partial Resubstitution

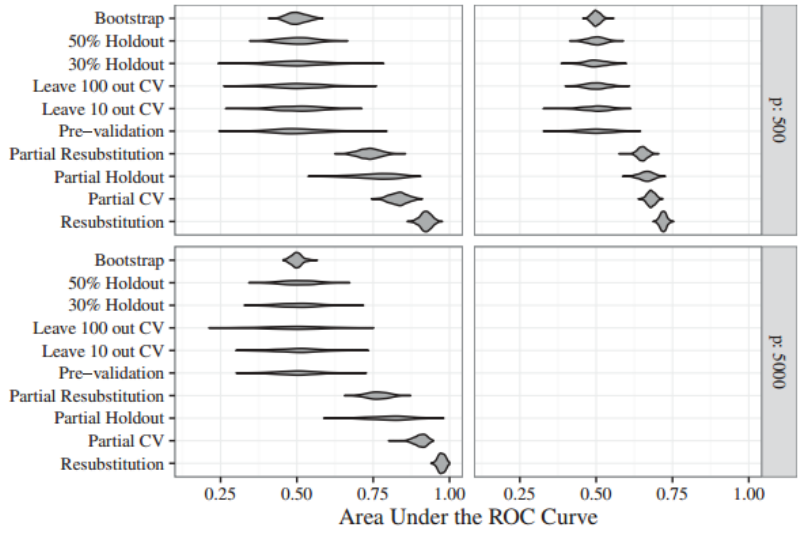
- Split into training and validation datasets
- Estimate the signature using the training dataset
- Report performance on combined training and validation datasets

BIASED!

Illustration



Illustration



Key principles

1. Valid estimates of performance obtained on data not used to develop the signature
2. Avoid leaking of information from the training to validation subsets
3. The **entire** model development process (variable selection, model fit, everything) must be kept separate from the validation data
4. Be skeptical of reports of extreme hazard ratios or perfect prediction, especially if the signature development process was complex

Signature development methods

How is the signature made?

- Include all features or just a subset?
  - Which subset
  - Identify features that cluster together
- How to combine the features?
  - Transformations
  - Weight and combine with regression
  - Estimating coefficients
  - Thresholds or cutoffs before or after combining

Key issues

Problem: Many variables relative to the number of samples (High dimensional data)

1. Data reduction, filtering
   - Select a subset of variables by screening them one-by-one or by some algorithm (e.g. stepwise)
   - Combine variables into a smaller set using clustering methods
2. Variable combination
   - Regularization: special model that includes a penalty for including too many features
     - Lasso, ridge regression, elastic net (Hastie et al.)
   - Standard multivariable regression
   - Tree based method, (e.g., CART, random forests)
3. Cutpoint identification (optional)
   - Choose threshold for making a decision

In practice

- Any combination of the above
- Explosion of the number of method developed in statistical literature
- *Ad hoc* methods used in clinical literature

Complex and convoluted methods make it difficult to ensure that the entire procedure is validated

Ensemble/Stacking methods

- Unless you generated the data, it is impossible to predict which prediction development algorithm will work best
- The principle of clean training-validation separation makes it hard to choose one method a priori
- instead, use them all and average the results!

Define a *library* of algorithms, each of which gives you a way to estimate a predictive model.

Superlearner

On a V-fold cross-validation partition of the data,

1. Fit each algorithm separately, on the training portion. Get V model fits for each algorithm
2. Obtain predictions for Y on the validation set for each fold, get predictions for each subject and algorithm
3. Compute the cross-validated performance for each algorithm
4. Regress the outcome Y on the set of cross validated predictions to obtain optimal weights
5. Fit each algorithm on the full data, combine with the optimal weights estimated in step 4.

Outperforms any individual algorithm in the library

Polley, Eric C. and van der Laan, Mark J., "Super Learner In Prediction" (May 2010). U.C. Berkeley Division of Biostatistics Working Paper Series. Working Paper 266. https://biostats.bepress.com/ucbbiostat/paper266

Stata commands?

- A few built in commands
- Many user-written modules

https://www.stata.com/stata-news/news33-4/users-corner/

Final Thoughts and Other topics

- Measures of association (e.g. hazard/odds ratio) are commonly reported to demonstrate that prediction models are good
  - Shows that there is some signal
  - Not good enough to evaluate utility of the predictions
- Survival outcomes present further challenges
- Methods for signature development
- Assay issues, can we measure the features or even the outcome, reliably?
- Hidden problems in the literature, poor reporting of methods
- Many predictive models are reported, very few are used in practice, why?