Three Aspects of Predictive Modeling

Max Kuhn, Ph.D

Pfizer Global R&D
Groton, CT
max.kuhn@pfizer.com

Outline

- "Predictive modeling" definition
- Some example applications
- A short overview and example
- How is this different from what statisticians already do?
- Unmet challenges in applied modeling
- Predictive models with big data
- Ethical considerations

“Predictive Modeling”

Define That!

Rather than saying that method X is a predictive model, I would say:

Predictive Modeling

is the process of creating a model whose *primary* goal is to achieve high levels of accuracy.

In other words, a situation where we are concerned with making the best possible prediction on an individual data instance.

(aka pattern recognition)(aka machine learning)

Models

So, in theory, a linear or logistic regression model is a predictive model?

Yes.

As will be emphasized during this talk:

- the quality of the prediction is the focus
- model interpretability and inferential capacity are not as important

Example Applications

Examples

- **spam detection**: we want the most accurate prediction that minimizes false positives and eliminates spam
- **plane delays**, **travel time**, etc.
- **customer volume**
- **sentiment analysis** of text
- **sale price of a property**

For example, does anyone care *why* an email or SMS is labeled as spam?

Quantitative Structure Activity Relationships (QSAR)

Pharmaceutical companies screen millions of molecules to see if they have good properties, such as:

- biologically potent
- safe
- soluble, permeable, drug–like, etc

We synthesize many molecules and run lab tests ("assays") to estimate the characteristics listed above.

When a medicinal chemist designs a new molecule, he/she would like a prediction to help assess whether we should synthesized it.

Medical Diagnosis

A patient might be concerned about having cancer on the basis of some on physiological condition their doctor has observed.

Any result based on imaging or a lab test should, above all, be as accurate as possible (more on this later).

If the test does indicate cancer, very few people would want to know if it was due to high levels of:

- human chorionic gonadotropin (HCG)
- alpha-fetoprotein (AFP) or
- lactate dehydrogenase (LDH)

Accuracy is the concern

Managing Customer Churn/Defection

When a customer has the potential to switch vendors (e.g. phone contract ends), we might want to estimate

- the probability that they will churn
- the expected monetary loss from churn

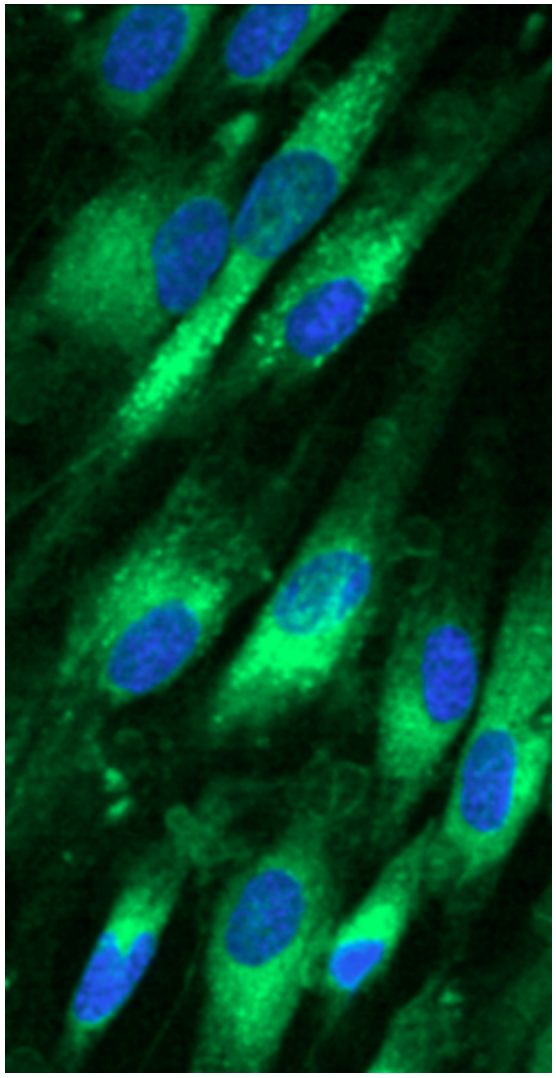
Based on these quantities, you may want to offer a customer an *incentive* to stay.

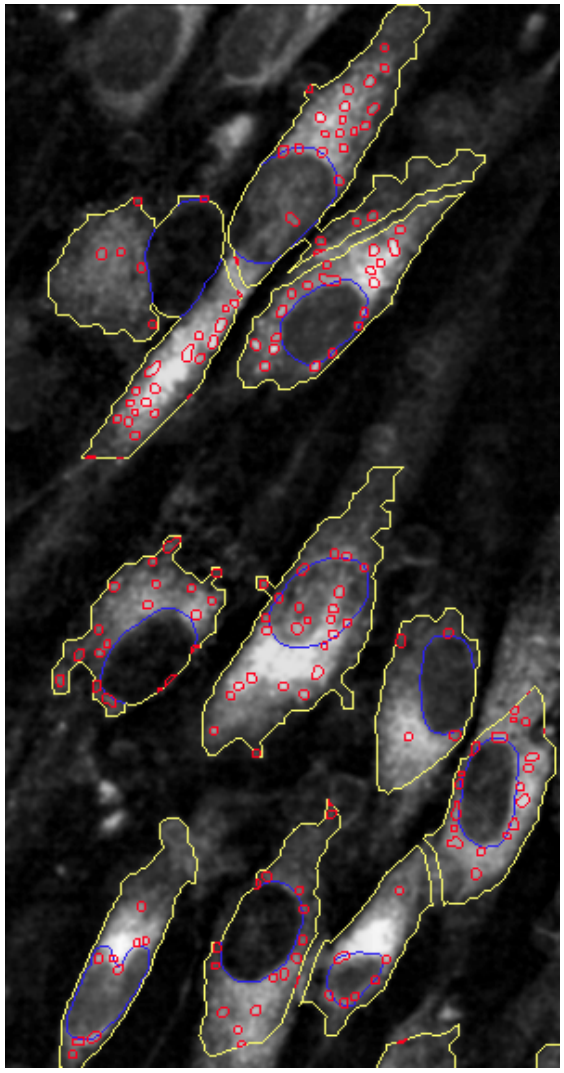
Poor predictions will result in loss of revenue from a customer loss or from a inappropriately applied incentive.

The goal is to minimize financial loss at the individual customer level

An Overview of the Modeling Process

Cell Segmentation





Cell Segmentation

Individual cell results are aggregated so that decisions can be made about specific compounds

Improperly segmented objects might compromise the quality of the data so an algorithmic filter is needed.

In this application, we have measurements on the size, intensity, shape or several parts of the cell (e.g. nucleus, cell body, cytoskeleton).

Can these measurements be used to predict poorly segmentation using a set of manually labeled cells?

The Data

Hill et al (2007) scored 2019 cells into these two bins: well–segmented (`WS`) or poorly–segmented (`PS`).

There are 58 measurements in each cell that can be used as predictors.

The data are in the caret package.

Model Building Steps

Common steps during model building are:

- estimating model parameters (i.e. training models)
- determining the values of tuning parameters that cannot be directly calculated from the data
- calculating the performance of the final model that will generalize to new data

Model Building Steps

How do we "spend" the data to find an optimal model? We typically split data into training and test data sets:

- **Training Set**: these data are used to estimate model parameters and to pick the values of the complexity parameter(s) for the model.
- **Test Set** (aka validation set): these data can be used to get an independent assessment of model efficacy. They should not be used during model training.

Spending Our Data

The more data we spend, the better estimates we'll get (provided the data is accurate). Given a fixed amount of data,

- too much spent in training won't allow us to get a good assessment of predictive performance. We may find a model that fits the training data very well, but is not generalizable (over–fitting)
- too much spent in testing won't allow us to get good estimates of model parameters

Spending Our Data

Statistically, the best course of action would be to use all the data for model building and use statistical methods to get good estimates of error.

From a non–statistical perspective, many consumers of of these models emphasize the need for an untouched set of samples the evaluate performance.

The authors designated a training set ($n = 1009$) and a test set ($n = 1010$).

Over–Fitting

Over–fitting occurs when a model inappropriately picks up on trends in the training set that do not generalize to new samples.

When this occurs, assessments of the model based on the training set can show good performance that does not reproduce in future samples.

Some models have specific "knobs" to control over-fitting

- neighborhood size in nearest neighbor models is an example
- the number if splits in a tree model

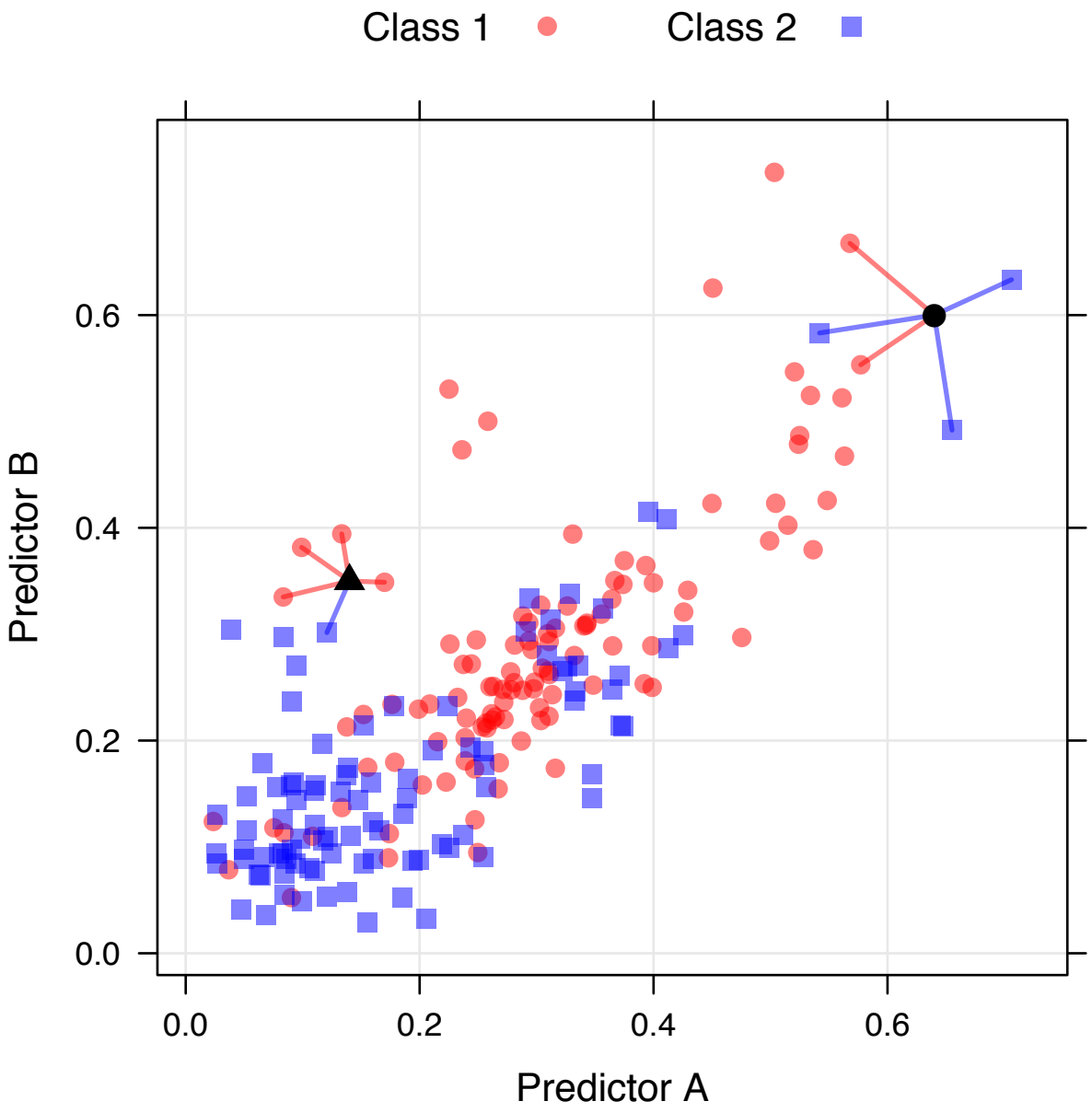
Over–Fitting

Often, poor choices for these parameters can result in over-fitting

For example, the next slide shows a data set with two predictors. We want to be able to produce a line (i.e. decision boundary) that differentiates two classes of data.

Two new points are to be predicted. A 5–nearest neighbor model is illustrated.

K–Nearest Neighbors Classification



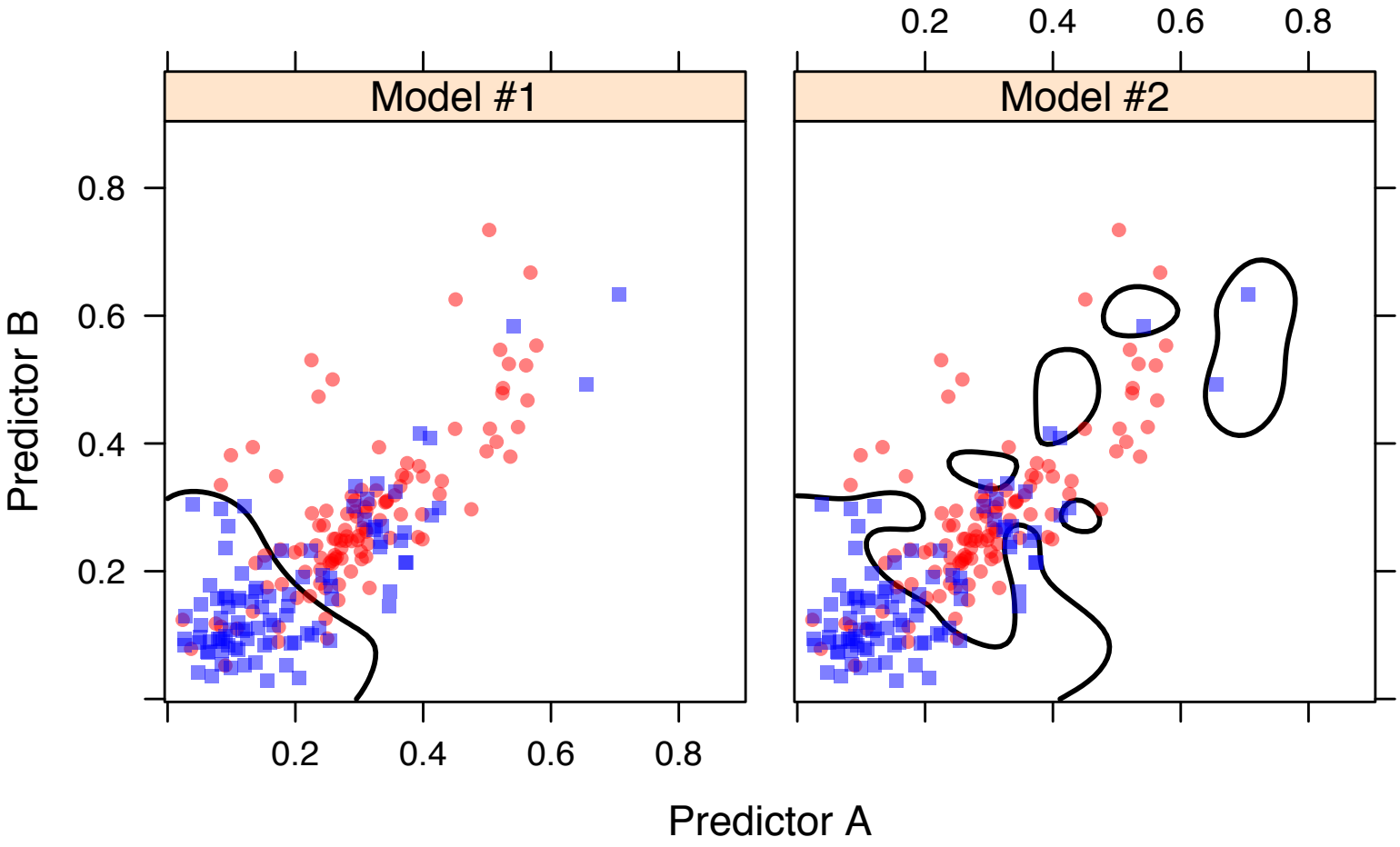
Over–Fitting

On the next slide, two classification boundaries are shown for the a different model type not yet discussed.

The difference in the two panels is solely due to different choices in tuning parameters.

One over–fits the training data.

Two Model Fits



Characterizing Over–Fitting Using the Training Set

One obvious way to detect over–fitting is to use a test set. However, repeated "looks" at the test set can also lead to over–fitting

Resampling the training samples allows us to know when we are making poor choices for the values of these parameters (the test set is not used).

Examples are cross–validation (in many varieties) and the bootstrap.

These procedures repeated split the *training data* into subsets used for modeling and performance evaluation.

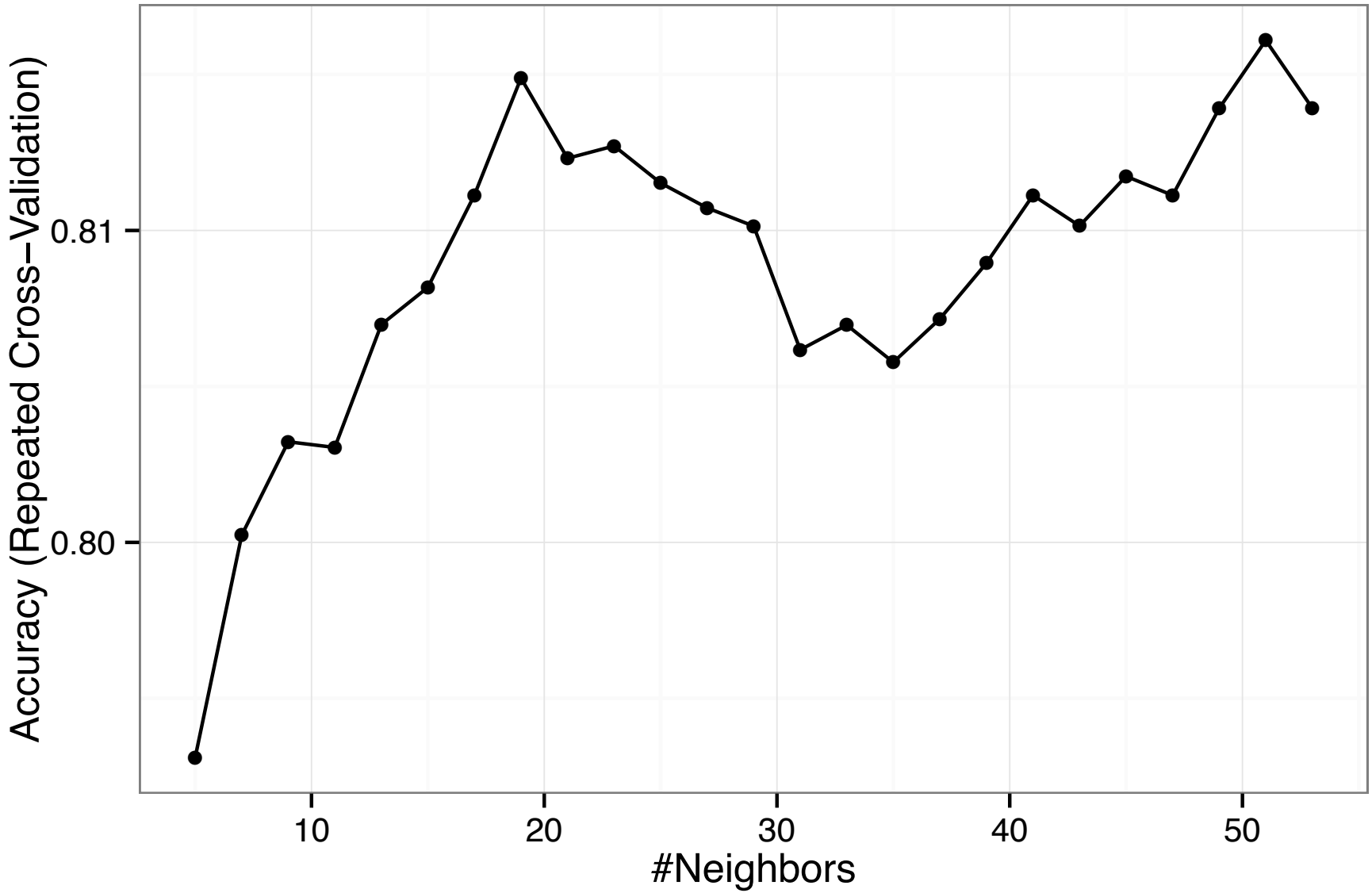
The Big Picture

We think that resampling will give us honest estimates of future performance, but there is still the issue of which sub–model to select (e.g. 5 or 10 NN).

One algorithm to select sub–models:

```
Define sets of model parameter values to evaluate;
for each parameter set do
    for each resampling iteration do
        Hold–out specific samples ;
        Fit the model on the remainder;
        Predict the hold–out samples;
    end
    Calculate the average performance across hold–out predictions
end
Determine the optimal parameter value;
Create final model with entire training set and optimal parameter
value;
```

K–Nearest Neighbors Tuning



Next Steps

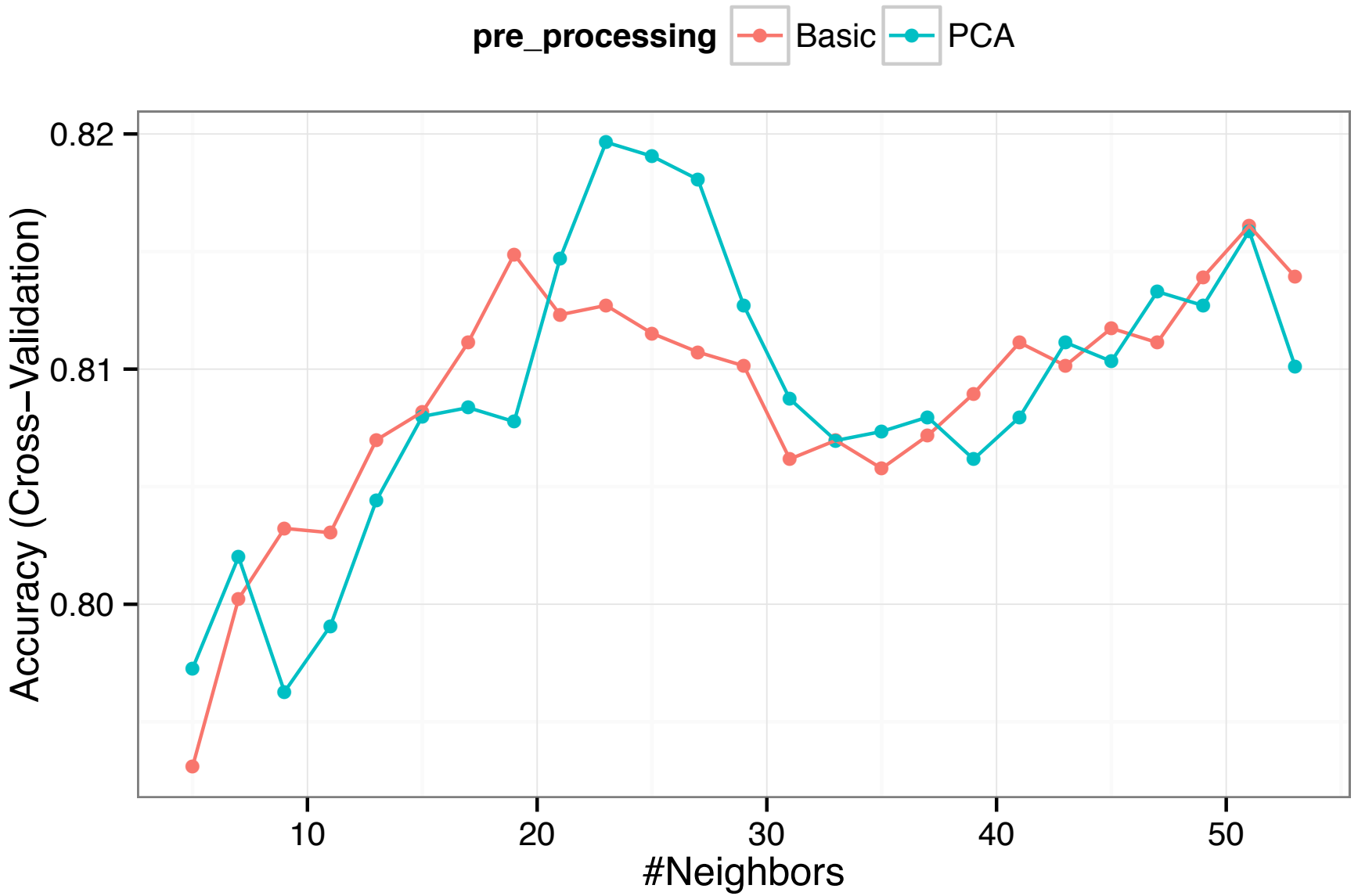
Normally, we would try different approaches to improving performance:

- different models,
- other pre–processing techniques,
- model ensembles, and/or
- feature selection (i.e. variable selection)

For example, the degree of correlation between the predictors in these data is fairly high. This may have had a negative impact on the model.

We can fit another model that uses a sufficient number of principal components in the K–nearest neighbor model (instead of the original predictors).

Pre-Processing Comparison



Evaluating the Final Model Candidate

Suppose we evaluated a panel of models and the cross–validation results indicated that the K–nearest neighbor model using PCA had the best cross–validation results.

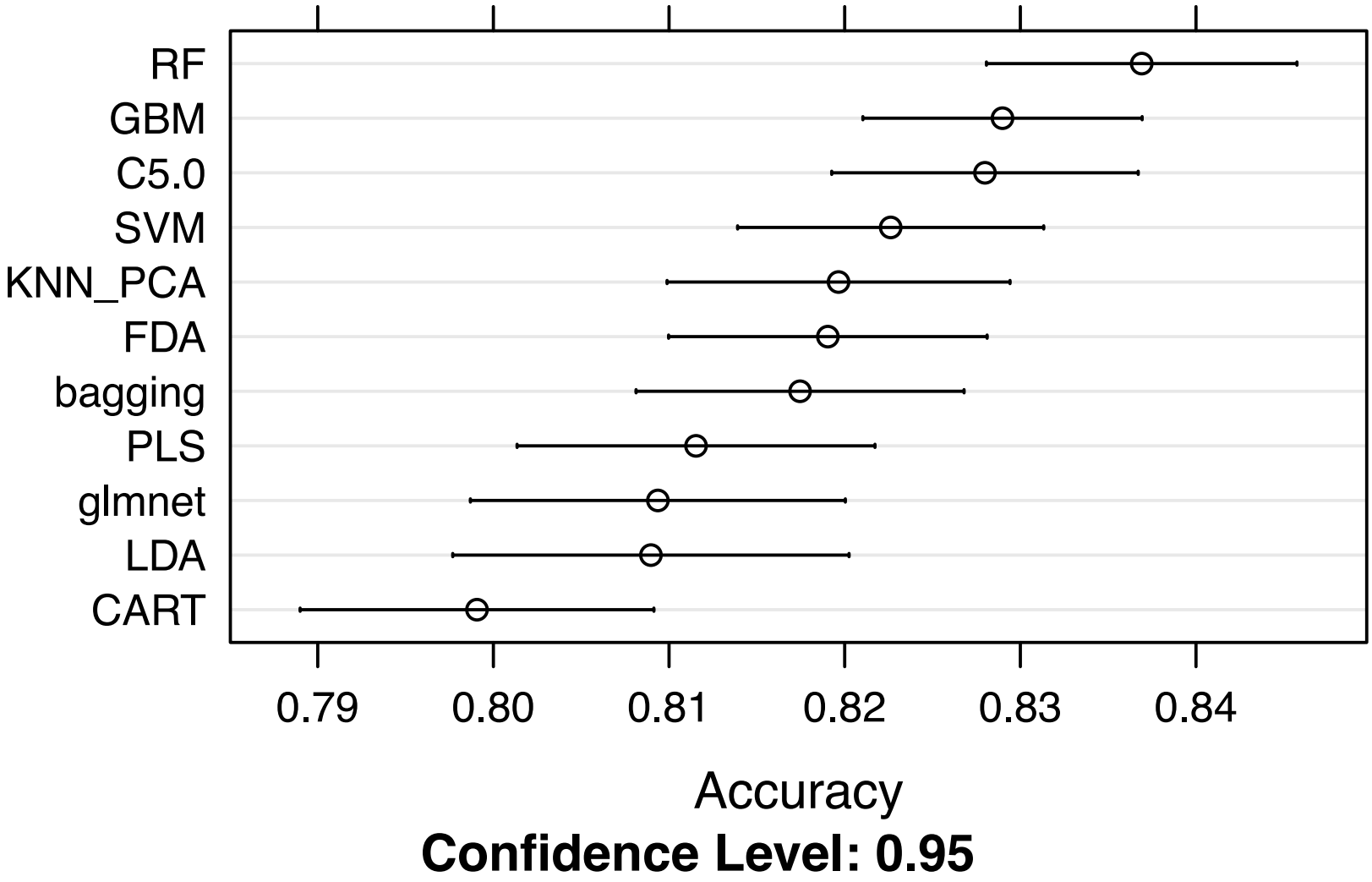
We would use the test set to verify that there were no methodological errors.

The test set accuracy was 80.7%.

The CV accuracy was 82%.

Pretty Close!

Segmentation CV Results for Different Models



Comparisons to Traditional Statistical Analyses

Good References On This Topic

Breiman, L. (1997). No Bayesians in foxholes. *IEEE Expert*, 12(6), 21–24.

Breiman, L. (2001). Statistical modeling: The two cultures. *Statistical Science*, 16(3), 199–215. (plus discussants)

Ayres, I. (2008). *Super Crunchers: Why Thinking-By-Numbers is the New Way To Be Smart*, Bantam.

Shmueli, G. (2010). To explain or to predict? *Statistical Science*, 25(3), 289–310.

Boulesteix, A.-L., & Schmid, M. (2014). Machine learning versus statistical modeling. *Biometrical Journal*, 56(4), 588–593. (and other articles in this issue)

Empirical Validation

Previously there was a emphasis on empirical assessments of accuracy.

While accuracy isn't the best measure, metrics related to errors of some sort are way more important here than in traditional statistics.

Statistical criteria (e.g. lack of fit tests, p–values, likelihoods, etc.) are not directly related to performance.

Friedman (2001) describes an example related to boosted trees with MLE:

> *"[...] degrading the likelihood by overfitting actually* improves *misclassification error rates. Although perhaps counterintuitive, this is not a contradiction; likelihood and error rate measure different aspects of fit quality."*

Empirical Validation

Even some measure that are asymptotically related to error rates (e.g. AIC, BIC) are still insufficient.

Also, a number of statistical criteria (e.g. adjusted R^2) require degrees of freedom. In many predictive models, these do not exist or are much larger than the training set size.

Finally, there is often an interest in measuring *nonstandard loss functions* and optimizing models on this basis.

For the customer churn example, the loss of revenue related to false negatives and false positives may be different. The loss function may not fit nicely into standard decision theory so evidence–based assessments are important.

Statistical Formalism

Traditional statistics are often used for making inferential statements regarding parameters or contrasts of parameters.

For this reason, there is a fixation on the appropriateness of the models related to

- necessary distributional assumptions required for theory
- relatively simple model structure to keep the math trackable
- the sanctity of degrees of freedom.

This is critical to make appropriate inferences on model parameters.

However, letting these go allows the user greater flexibility to increase accuracy.

Examples

- complex or nonlinear pre–processing (e.g. PCA, spatial sign)
- ensembles of models (boosting, bagging, random forests)
- overparameterized but highly regularized nonlinear models (SVN, NNets)
- Quinlan (1993) describing pessimistic pruning, notes that it:

  "does violence to statistical notions of sampling and confidence limits, so the reasoning should be taken with a grain of salt."

- Breiman (1997) notes that there were "No Bayesians in foxholes."

Three Sweeping Generalizations About Models

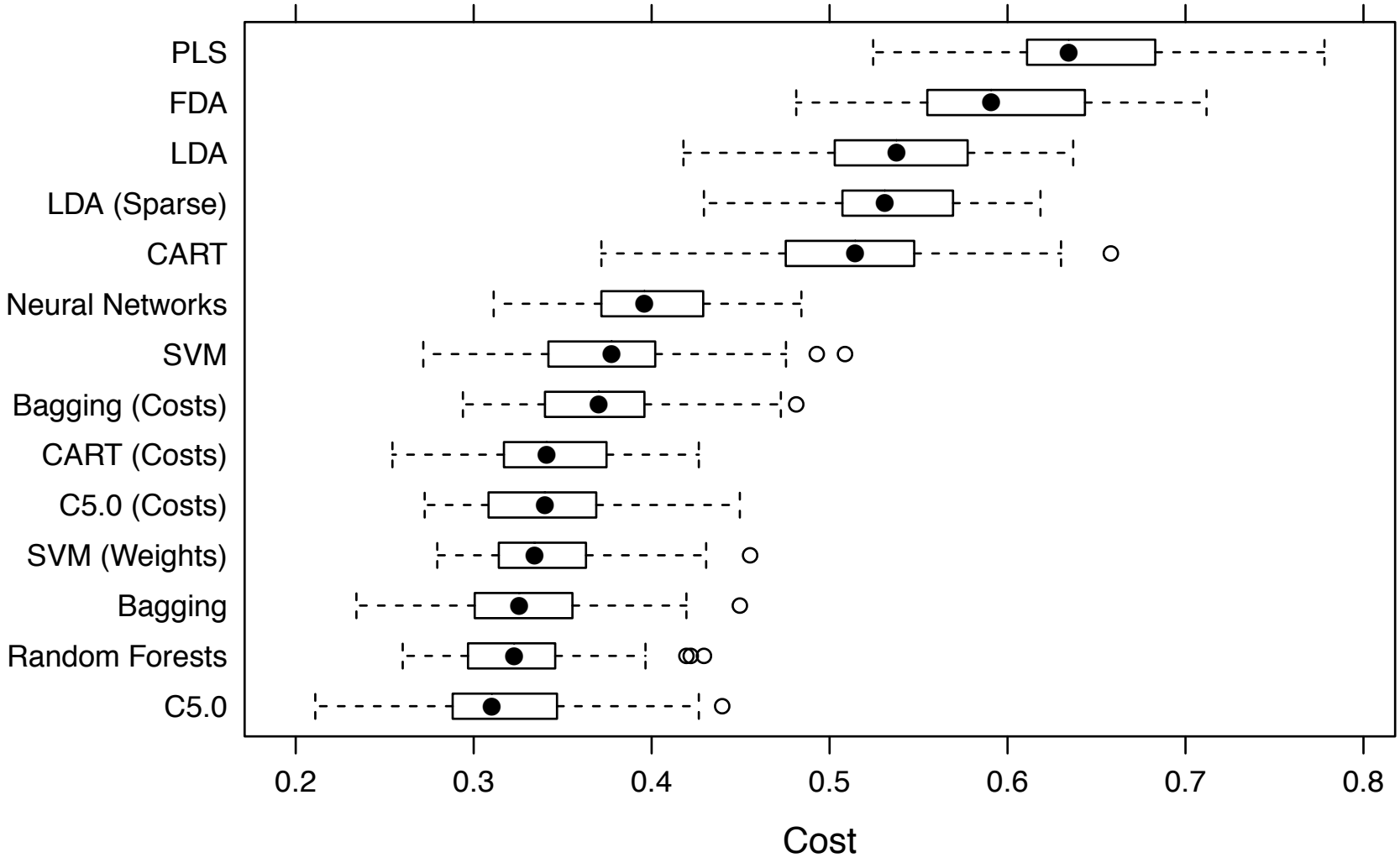
① In the absence of any knowledge about the prediction problem, no model can be said to be uniformly better than any other (No Free Lunch theorem)

② There is an inverse relationship between model accuracy and model interpretability

③ Statistical validity of a model is not always connected to model accuracy

Cross–Validation Cost Estimates

(HPC Case Study, APM, pg 454)



Unmet Challenges in Applied Modeling

Diminishing Returns on Models

Honestly, I think that we have plenty of effective models.

Very few problems are solved by inventing yet another predictive model.

Exceptions:

- using unlabeled data
- severe class imbalances
- applicability domain techniques and prediction confidence assessments

It is more important to improve the **features** that go into the model.

"Feature Engineering"

For example, I have a data set of daily measurements on public transit and one possible predictor is the date.

How should we encode this?

- day of the month, day of the year?
- week? month? year? (numeric or categorical?)
- day of the week?
- is it a holiday?
- is it during the school semester?

Feature engineering try to enter the predictor(s) into a model in a way that maximizes the benefit to the model.

"Feature Engineering"

Some algorithmic feature engineering algorithms exist such as MARS, autoencoders, PCA, ICA, NNMF, etc.

GAMs can also be useful to elucidate non–linear relationships between an outcome and a predictor.

Better interaction and sub–space detection techniques would also be a big help.

The key is to have an algorithmic method for reducing bias in low-complexity model without selection bias or overfitting.

Ethical Considerations

Ethical Considerations

There are some situations where using a sub–optimal model can be *unethical*.

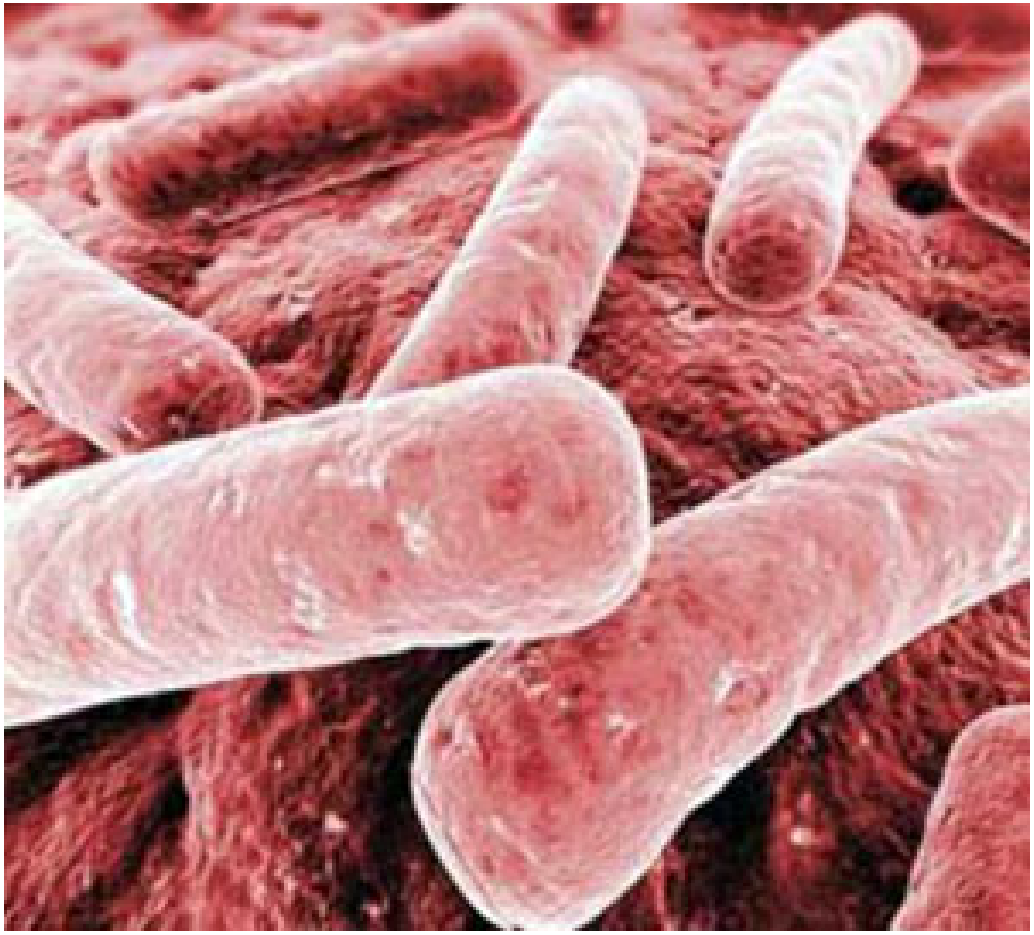
My experiences in molecular diagnostics gave me some perspective on the friction point between pure performance and notions about validity.

Another generalization: doctors and regulators are apprehensive about black–box models since they cannot make sense of it and assume that the validation of these models is lacking.

I had an experience as a "end user" of a model (via a lab test) when I was developing algorithms for a molecular diagnostic company.

Antimicrobial Susceptibility Testing

Klebsiella pneumoniae



klebsiella-pneumoniae.org/


St. Joseph
Medical Center

Predictive Models with Big Data

What About Big Data?

The advantages and issues related to Big Data can be broken down into two areas:

- Big N: more samples or cases
- Big P: more variables or attributes or fields

Mostly, I think the catchphrase is associated more with N than P.

Does Big Data solve my problems?

Maybe[1]

[1] the basic answer given by every statistician throughout time

Can You Be More Specific?

It depends on

- what are you using it for?
- does it solve some *unmet need*?
- does it get in the way?

Basically, it comes down to:

$$\text{Bigger Data} \neq \text{Better Data}$$

at least not necessarily.

Big N - Interpolation

One situation where it probably doesn't help is when samples are added *within the mainstream of the data*

In effect, we are just filling in the predictor space by increasing the granularity.

After the first 10,000 or so observations, the model will not change very much.

This does pose an interesting interaction within the **variance–bias trade off**.

Big N goes a long way to reducing the model variance. Given this, can high variance/low bias models be improved?

High Variance Model with Big N

Maybe.

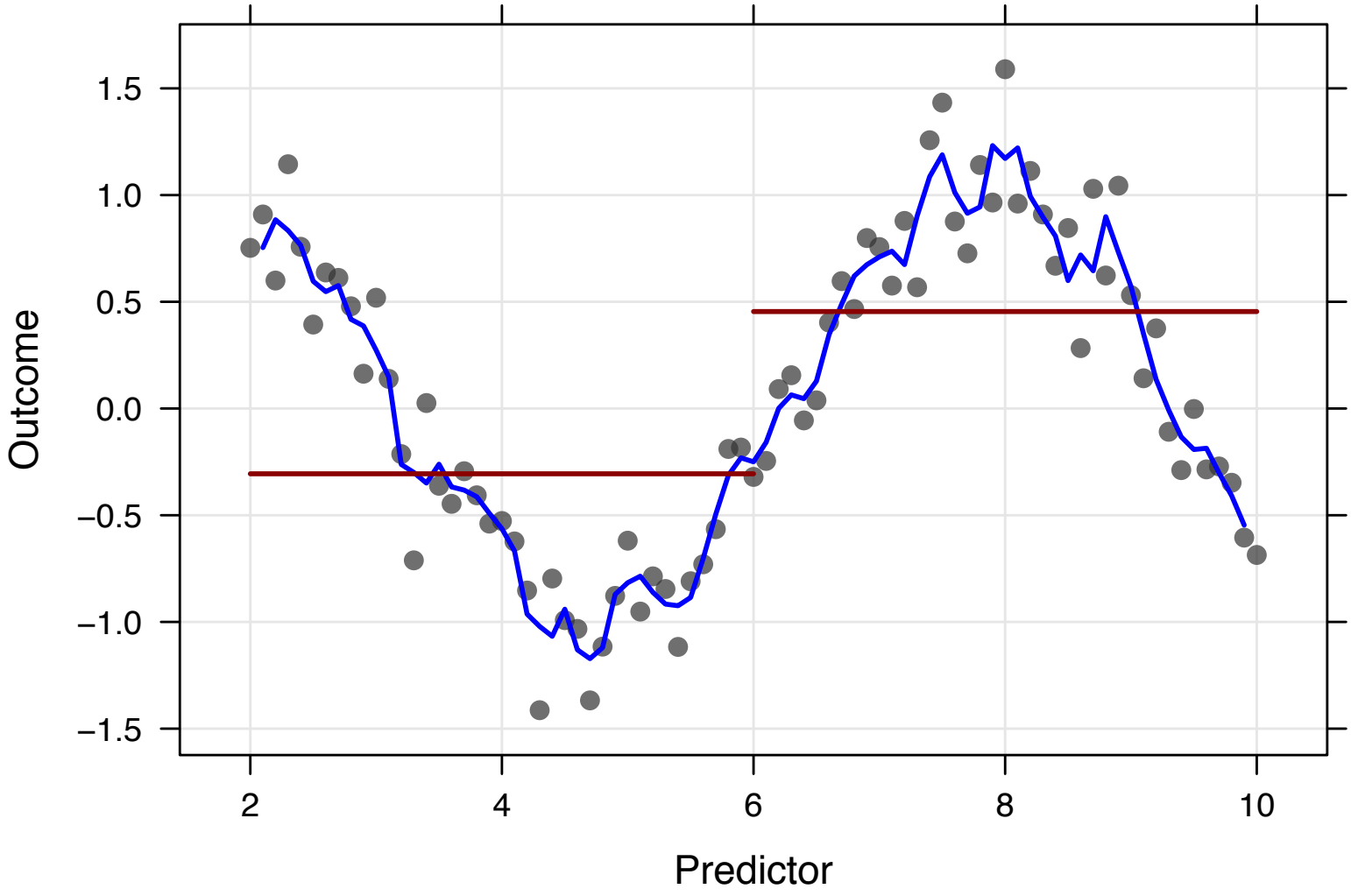
Many high(ish) variance/low bias models (e.g. trees, neural networks, etc.) tend to be very complex and computationally demanding.

Adding more data allows these models to more accurately reflect the complexity of the data but would require specialized solutions to be feasible.

At this point, the best approach is supplanted by the available approaches (not good).

Form should still follow function.

Model Variance and Bias



Low Variance Model with Big N

What about low variance/high bias models (e.g. logistic and linear regression)?

There is some room here for improvement since the abundance of data allows more opportunities for exploratory data analysis to tease apart the functional forms to lower the bias (i.e. improved *feature engineering*) or to select features.

For example, non–linear terms for logistic regression models can be parametrically formalized based on the results of spline or loess smoothers.

Diminishing Returns on N^*

At some point, adding more (* of the same) data does not do any good.

Performance stabilizes but computational complexity increases.

The modeler becomes hand–cuffed to whatever technology is available to handle large amounts of data.

Determining what data to use is more important than worrying about the technology required to fit the model.

Global Versus Local Models in QSAR

When developing a compound, once a "hit" is obtained, the chemists begin to tweak the structure to make improvements. The leads to a *chemical series*

We could build QSAR models on the large number of existing compounds (a global model) or on the series of interest (a local model).

Our experience is that local models beat global models the majority of the time.

Here, fewer (of the most relevant) compounds are better.

Our first inclination is to use all the data because our (overall) error rate should get better.

Like politics, *All Problems Are Local*.

Data Quality (QSAR Again)

One "Tier 1" screen is an assay for logP (the partition coefficient) which we use as a measure of "greasyness".

Tier 1 means that logP is estimated for most compounds (via model and/or assay)

There was an existing, high–throughput assay on a large number of historical compounds. However, the data quality was poor.

Several years ago, a new assay was developed that was lower throughput and higher quality. This became the default assay for logP.

The model was re-built on a small ($N \approx 1,000$) set chemically diverse compounds.

In the end, fewer compounds were assayed but the model performance was much better and costs were lowered.

Big N and/or P - Reducing Extrapolation

However, Big N might start to sample from rare populations.

For example:

- a customer cluster that is less frequent but has high profitability (via Big N).
- a specific mutation or polymorphism (i.e. a rare event) that helps derive a new drug target (via Big N *and* Big P)
- highly non–linear "activity cliffs" in computational chemistry can be elucidated (via Big N)

Now, we have the ability to solve some unmet need.

Thanks!

Backup Slides

How Does Statistics Contribute to Predictive Modeling?

Many statistical principals still apply and have greatly improved the field:

- resampling
- the variance–bias tradeoff
- sampling bias in big data
- parsimony principal (AOTBE, keep the simpler model)
- L_1 and/or L_2 regularization

Statistical improvements to boosting are a great example of this.

What Can Drive Choice of Methodology?

Important Considerations

It is fairly common to have a number of different models with equivalent levels of performance.

If a predictive model will be deployed, there are a few practical considerations that might drive the choice of model.

- if the prediction equation is to be numerically optimized, models with smooth functions might be preferable.
- the characteristics of the data (e.g. multicollinearity, imbalanced, etc) will often constrain the feasible set of models.
- large amounts of unlabeled data can be exploited by some models

Ease of Use/Model Development

If a large number of models will be created, using a *low maintenance* technique is probably a good idea.

QSAR is a good example. Most companies maintain a large number of models for different endpoints. These models are constantly updated too.

Trees (and their ensembles) tend to be low maintenance; neural networks are not.

However, if the model must have the absolute best possible error rate, you might be willing to put a lot of work into the model (e.g. deep neural networks)

How will the model be deployed?

If the prediction equation needs to be encoded into software or a database, concise equations have the advantage.

For example, random forests require a large number of unpruned trees. Even for a reasonable data set, the total forest can have thousands of terminal nodes (39,437 if/then statements for the cell model)

Bagged trees require dozens of trees and might yield the same level of performance but with a smaller footprint

Some models (K-NN, support vector machines) can require storage of the original training set, which may be infeasible.

Neural networks, MARS and a few other models can yield compact nonlinear prediction equations.