Predictive Models in Health Research

Trevor Hastie
Department of Statistics
Department of Biomedical Data Science
Stanford University

41st Meeting of the International Society for Clinical Biostatistics
ISCB, Krakow, Poland, August 23-27, 2020
Some Take Home Messages

This talk is about supervised learning: building models from data that predict an outcome using a collection of input features.

• There are some powerful and exciting tools for making predictions from data.
• They are not magic! You should be skeptical. They require good data and proper internal validation.
• Human judgement and ingenuity are essential for their success.
• They may be overkill — if simpler and more traditional methods do as well, they are preferable (Occam’s razor).
Some Definitions

**Machine Learning** constructs algorithms that can learn from data.

**Statistical Learning** is a branch of applied statistics that emerged in response to machine learning, emphasizing statistical models and assessment of uncertainty.

**Data Science** is the extraction of knowledge from data, using ideas from mathematics, statistics, machine learning, computer science, engineering, ...

All of these are very similar — with different emphases.
Internal Model Validation

• **IMPORTANT!** Don’t trust me or anyone who says they have a wonderful machine learning algorithm, unless you see the results of a careful internal validation.

• Eg: divide data into two parts $A$ and $B$. Run algorithm on part $A$ and then test it on part $B$. **Algorithm must not have seen any of the data in part $B$.**

• If it works in part $B$, you have (some) confidence in it.
Internal Model Validation

- **IMPORTANT!** Don’t trust me or anyone who says they have a wonderful machine learning algorithm, unless you see the results of a careful internal validation.
- Eg: divide data into two parts $A$ and $B$. Run algorithm on part $A$ and then test it on part $B$. Algorithm must not have seen any of the data in part $B$.
- If it works in part $B$, you have (some) confidence in it.

Simple? Yes
Internal Model Validation

- **IMPORTANT!** Don’t trust me or anyone who says they have a wonderful machine learning algorithm, unless you see the results of a careful internal validation.

- Eg: divide data into two parts $A$ and $B$. Run algorithm on part $A$ and then test it on part $B$. Algorithm must not have seen any of the data in part $B$.

- If it works in part $B$, you have (some) confidence in it

  Simple? Yes
  Done properly in practice? Rarely

In God we trust. All others bring data. ∗∗

Statistical "proverb" sometimes attributed to W. Edwards Deming.
Internal Model Validation

- **IMPORTANT!** Don’t trust me or anyone who says they have a wonderful machine learning algorithm, unless you see the results of a careful internal validation.

- Eg: divide data into two parts $A$ and $B$. Run algorithm on part $A$ and then test it on part $B$. Algorithm must not have seen any of the data in part $B$.

- If it works in part $B$, you have (some) confidence in it. Simple? Yes. Done properly in practice? Rarely.

*In God we trust. All others bring data.*
Internal Model Validation

- **IMPORTANT!** Don’t trust me or anyone who says they have a wonderful machine learning algorithm, unless you see the results of a careful internal validation.

- Eg: divide data into two parts $A$ and $B$. Run algorithm on part $A$ and then test it on part $B$. Algorithm must not have seen any of the data in part $B$.

- If it works in part $B$, you have (some) confidence in it

  Simple? Yes
  Done properly in practice? Rarely

  *In God we trust. All others bring data.*

* Statistical “proverb” sometimes attributed to W. Edwards Deming.*
Big data vary in *shape*. These call for different approaches.

**Wide Data**
- Thousands / Millions of Variables
- Hundreds of Samples
  - Screening and fdr,
  - Lasso, SVM, Stepwise
  - We have too many variables; prone to overfitting.
  - Need to remove variables, or regularize, or both.

**Tall Data**
- Tens / Hundreds of Variables
- Thousands / Millions of Samples
  - GLM, Random Forests,
  - Boosting, Deep Learning
  - Sometimes simple models (linear) don’t suffice.
  - We have enough samples to fit nonlinear models with many interactions, and not too many variables.
  - Good automatic methods for doing this.
Big data vary in *shape*. These call for different approaches.

**Tall and Wide Data**
- Thousands / Millions of Variables
- Millions to Billions of Samples

**Tricks of the Trade**
- Exploit sparsity
- Random projections / hashing
- Variable screening
- Subsample rows
- Divide and recombine
- Case/ control sampling
- MapReduce
- ADMM (divide and conquer)
• Large scale lasso models using \texttt{glmnet} (tall and wide)
• \textbf{Random Forest} used as ensemble learner (tall)
• \textbf{Deep learning} used for image classification (wide)
• Machine learning overkill (very wide)
glmnet

Fit regularization paths for a variety of GLMs with lasso and elastic net penalties; e.g. logistic regression

\[
\log \frac{\Pr(Y = 1 \mid X = x)}{\Pr(Y = 0 \mid X = x)} = \beta_0 + \sum_{j=1}^{p} x_j \beta_j
\]

- Lasso penalty [Tibshirani, 1996] induces sparsity in coefficients: \( \sum_{j=1}^{p} |\beta_j| \leq s \). It shrinks them toward zero, and sets many to zero.
- Fit efficiently using coordinate descent. Handles sparse \( X \) naturally, and exploits sparsity of solutions, warms starts, variable screening, and includes methods for model selection using cross-validation.

*glmnet team: TH, Jerome Friedman, Rob Tibshirani, Noah Simon, Junyang Qian, Balasubramanian Narasimhan, Ken Tay.*
A Fast and Scalable Framework for Large-scale and Ultrahigh-dimensional Sparse Regression with Application to the UK Biobank

Jinyang Qian, Yosuke Tanigawa, Wenfei Du, Matthew Aguirre, Chris Chang, Robert Tibshirani, Manuel A. Rivas, Trevor Hastie

doi: https://doi.org/10.1101/630079

This article is a preprint and has not been certified by peer review [what does this mean?].

Abstract

The UK Biobank (Bycroft et al., 2018) is a very large, prospective population-based cohort study across the United Kingdom. It provides unprecedented opportunities for researchers to investigate the relationship between genotypic information and phenotypes of interest. Multiple regression methods, compared with GWAS, have already been showed to greatly improve the prediction performance for a variety of phenotypes. In the high-dimensional settings, the lasso (Tibshirani, 1996), since its
Build phenotype prediction models using genotypes at 800K loci (SNPs) for 500K individuals. Here we predict Standing Height using lasso model via glmnet, and achieve $R^2 = 70\%$. 

---

UK Biobank Genomic Data

![Graph showing R² values for different models across various lambda indices.](image)
Algorithmic details in pictures

Figure 1: The lasso coefficient profile that shows the progression of the BASIL algorithm. The previously finished part of the path is colored grey, the newly completed and verified is in green, and the part that is newly computed but failed the verification is colored red.

of expensive disk read operations. At each iteration, we roll out the solution path progressively, which is illustrated in Figure 1 and will be detailed in the next section. In addition, we propose optimization specific for the SNP data in the UK Biobank studies to speed up the procedure.

1.4 Outline of the paper

The rest of the paper is organized as follows.

• Section 2 describes the proposed batch screening iterative lasso (BASIL) algorithm for the Gaussian family in detail and its extension to other problems such as logistic regression.

• Section 3 discusses related methods and packages for solving large-scale lasso problems.
Build polygenic risk scores for disease phenotypes
Note: only 6347 variants in active path for Lasso are included in the coefficient plot.
<table>
<thead>
<tr>
<th>Variable</th>
<th>Level</th>
<th>N</th>
<th>Events</th>
<th>Hazard Ratio</th>
<th>P value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polygenic risk score</td>
<td>40-60%</td>
<td>62139</td>
<td>5017</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bottom 10%</td>
<td>31036</td>
<td>896</td>
<td>0.36 (0.34, 0.39)</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td></td>
<td>Top 1%</td>
<td>3085</td>
<td>1494</td>
<td>7.31 (6.90, 7.74)</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td></td>
<td>Top 10%</td>
<td>15480</td>
<td>3636</td>
<td>3.06 (2.93, 3.20)</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td></td>
<td>Top 5%</td>
<td>12403</td>
<td>3894</td>
<td>4.27 (4.09, 4.45)</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Eosinophil threshold</td>
<td>below .18</td>
<td>73655</td>
<td>6392</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>above .18</td>
<td>50488</td>
<td>8545</td>
<td>1.61 (1.56, 1.67)</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Sex</td>
<td>Female</td>
<td>66292</td>
<td>8330</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Male</td>
<td>57851</td>
<td>6607</td>
<td>0.87 (0.85, 0.90)</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Body Mass Index</td>
<td>Below 25</td>
<td>40969</td>
<td>4414</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>25-30</td>
<td>53273</td>
<td>6220</td>
<td>1.04 (1.00, 1.08)</td>
<td>0.06</td>
</tr>
<tr>
<td></td>
<td>30-35</td>
<td>21600</td>
<td>2880</td>
<td>1.12 (1.07, 1.17)</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td></td>
<td>35-40</td>
<td>6010</td>
<td>954</td>
<td>1.26 (1.17, 1.35)</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td></td>
<td>Above 40</td>
<td>2291</td>
<td>469</td>
<td>1.59 (1.44, 1.75)</td>
<td>&lt;0.001</td>
</tr>
</tbody>
</table>
More on glmnet

Current version 4.02. Check out the website at glmnet.stanford.edu, and find the four vignettes there.

- All GLM family() choices are accommodated, along with the original hardwired families. e.g.
  family=binomial(link="probit"),
  family=negative.binomial(theta=5)

- New relax=TRUE argument which allows for refitting the active set without regularization.

- C index for survival models

- New functions for assessing fits

- Convenient functions for fitting big GLMs

- Progress bar via trace=TRUE
Predicting the Pathogenicity of Missense Variants

Goal: prioritize list of candidate genes for prostate cancer

Joint work with Epidemiology colleagues Weiva Sieh, Nilah Monnier Ioannidis, Joe Rothstein, Alice Whittemore, · · ·

REVEL — rare exome variant ensemble learner

Approach

- A number of existing scores for disease status do not always agree (e.g. SIFT, MutPred).
- Idea is to use a Random Forest algorithm to integrate these scores into a single consensus score for predicting disease.
- We will use existing functional prediction scores, conservation scores, etc as features — 12 features in all.
- Data acquired through Human Gene Mutation Database, SwissVar and ClinVar.

<table>
<thead>
<tr>
<th></th>
<th>Neutral</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>123,706</td>
<td>6,182</td>
</tr>
<tr>
<td>Test</td>
<td>2,406</td>
<td>1,953</td>
</tr>
</tbody>
</table>
Correlation of Features
Trees use the features to create subgroups in the data to refine the estimate of disease. Shallow trees are too coarse/inaccurate.
Random Forests

Leo Breiman (1928–2005)

- Deep trees (fine subgroups) are more accurate, but very noisy.
- Idea: fit many (1000s) different and very-deep trees, and average their predictions to reduce the noise.
- How to get different trees?
  - Grow trees to bootstrap subsampled versions of the data.
  - Randomly ignore variables as candidates for splits.

Random Forests are very effective and give accurate predictions. They are automatic, and give good CV estimates of prediction error (for free!). R package RandomForest.
Results for REVEL

Performance evaluated on independent test set, and REVEL compared with 7 other ensemble competitors.
Feature Importance

Relative importance

- FATHMM
- VEST
- MutationAssessor
- MutPred
- Polyphen2 HVAR
- PROVEAN
- phyloP (vertebrate)
- Polyphen2 HDIV
- SiPhy
- LRT
- GERP++ RS
- phyloP (placental)
- phastCons (primate)
- phastCons (placental)
- MutationTaster
- SIFT
- FATHMM
- VEST
- PROVEAN
- phastCons (vertebrate)
- Polyphen2 HVAR
- PROVEAN
- MutPred
- MutationTaster
- SIFT
- phastCons (primate)
- phastCons (vertebrate)
- phastCons (primate)
- phastCons (placental)
- phastCons (placental)
- phastCons (placental)
- phastCons (placental)
- phastCons (placental)
Convolutional neural network (CNN) for image modeling.

- Big successes with image classification. Revolutionary!
- Many digital images in healthcare: mammograms, MRI scans, X-rays, optical scans, skin photographs, ...
- Modern software tools accessible to all (e.g. keras and tensorflow in R).
Diabetic Retinopathy

Patient without DR

Patient with DR

10, 12 DR Absent
14, 15, 20 DR Questionable
35 Mild NPDR
43 Moderate NPDR
47 Moderately Severe NPDR
53 Severe NPDR

60, 61 Mild PDR
65 Moderate PDR
71 High-risk PDR
75 High-risk PDR
81 Advanced PDR
85 Advanced PDR
Two recent articles (2019-20)

ARTICLE
Deep learning algorithm predicts diabetic retinopathy progression in individual patients
Filippo Arcadu¹,², Fethallah Benmansour¹,², Andreas Maunz¹,², Jeff Willis³,⁴, Zdenka Haskova³,⁴,⁷*, and Marco Prunotto¹,²,⁵,⁶,⁷*

JAMA | Original Investigation | INNOVATIONS IN HEALTH CARE DELIVERY
Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs
Varun Gulshan, PhD; Lily Peng, MD, PhD; Marc Coram, PhD; Martin C. Stumpe, PhD; Derek Wu, BS; Arunachalam Narayanaswamy, PhD; Subhashini Venugopalan, MS; Kasumi Widner, MS; Tom Madams, MEng; Jorge Cuadros, OD, PhD; Ramasamy Kim, OD, DNB; Rajiv Raman, MS, DNB; Philip C. Nelson, BS; Jessica L. Mega, MD, MPH; Dale R. Webster, PhD
Validation Set Performance for Referable Diabetic Retinopathy

[A] 8788 images, 8 experts [B] 1745 images, 7 experts
Image Segmentation

Deep Learning CNN on MRI scans

Identify deep cervical spine extensors and measure muscle fat infiltration after whiplash injury

Adapt pretrained network

Compare with expert raters
Deep Learning Convolutional Neural Networks for the Automatic Quantification of Muscle Fat Infiltration Following Whiplash Injury

Kenneth A Weber 1, Andrew C Smith 2, Marie Wasielewski 3, Kamran Eghtesad 4, Pranav A Upadhyayula 4, Max Wintermark 5, Trevor J Hastie 6, Todd B Parrish 7, Sean Mackey 4, James M Elliott 3 8 9

Affiliations + expand

PMID: 31138878  PMCID: PMC6538618  DOI: 10.1038/s41598-019-44416-8

GT = Ground Truth (average of three raters)
CNN = Convolutional Neural Network
Overkill - article in Nature Medicine 2001

Simple linear method does as well, and does gene selection

*Elements of Statistical Learning, Section 18.2*
Summary

• We saw examples of Lasso, Random Forest, and Deep Learning.

• We saw an example where a Neural Networks was overkill, and a simpler model was sufficient.
Summary

- We saw examples of Lasso, Random Forest, and Deep Learning.
- We saw an example where a Neural Networks was overkill, and a simpler model was sufficient.

... and now for some cheap marketing ...
Summary

- We saw examples of **Lasso**, **Random Forest**, and **Deep Learning**.
- We saw an example where a **Neural Networks** was overkill, and a simpler model was sufficient.

... and now for some cheap marketing ...
Summary

• We saw examples of **Lasso**, **Random Forest**, and **Deep Learning**.

• We saw an example where a **Neural Networks** was overkill, and a simpler model was sufficient.

... and now for some cheap marketing ...

Thank you!