

# Statistics in the age of data science, issues you can and can not ignore

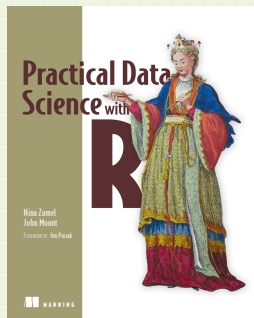
John Mount  
(data scientist, not a statistician)  
Win-Vector LLC  
<http://www.win-vector.com/>

These slides, all data and code: <http://winvector.github.io/DS/>

1

## Who I am

- John Mount
- Principal Consultant at Win-Vector LLC
  - Always looking for consulting, advising, training gigs
- One of the authors of Practical Data Science with R



2

## This talk

- Our most important data science tools are our theories and methods. Let us look a bit at their fundamentals.
- Large data gives the apparent luxury of wishing away some common model evaluation biases (instead of needing to apply the traditional statistical corrections).
- Conversely, to work agilely data scientists must act as if a few naive axioms of statistical inference were true (though they are not).
- I will point out some common statistical issues that do and do not remain critical problems when you are data rich.
- I will concentrate on the simple case of supervised learning.

3

## Outline

- (quick) What is data science?
- How can that work?
- An example critical task that gets easier when you have more data.
- What are some of the folk axioms of data science?
- How to design bad data.

4



4

## What is data science?

(please bear with me)

5



5

A term without meaning?

## What is Data Science: my position

- Data science is the continuation of data engineering and predictive analytics.
- More data allows *domain naive* models to perform well.
- Emphasis on prediction over harder statistical problems such as coefficient inference.
- Strong preference for easy procedures that become more statistically reliable as you accumulate more data.
- Reliance on strong black-box tools.

6



6

Can build deeper decision trees, introduce rarer indicators, and so on.

## Why does data science work at all?

7

Clearly we are ignoring some important domain science issues and statistical science issues, so how does data science work?

## Complicated domain theory doesn't *always* preclude easily observable statistical signals

8

"Maybe the molecule didn't go to graduate school."

(Will Welch defending the success of his approximate molecular screening algorithm, given he is a computer scientist and not a chemist.)



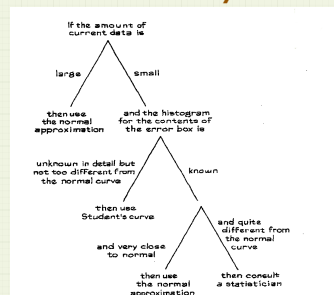
Example approximate docking (in this case using SUDE approximation, not Welch et al.'s Hammerhead).  
"Database Screening for HIV Protease Ligands: The Influence of Binding Site Conformation and Representation on Ligand Selectivity", Volker Schneck, Leslie A. Kuhn, Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology, Pages 242-251, AAAI Press, 1999.

<http://www.aaai.org/Papers/ISMB/1999/ISMB99-028.pdf>

You may not get the whole story, but you may not miss the whole story.

## A lot of deep statistics is about how to work correctly with small data sets

9



• From *Statistics 4th* edition, David Freedman, Robert Pisani, Roger Purves, Norton, 2007.

Ch. 26 page 493. Statistical efficiency is a huge worry when you don't have a lot of data.

10

What is a good example of a critical task that becomes easier when you have more data?

10



11

## Model assessment



- Estimating the performance of your predictive model on future instances
- A critical task
- Gets easier when you have more data:
  - Don't need to rely on statistical adjustments
  - Can reserve a single sample of data as a held-out test set (see "The Elements of Statistical Learning" 2nd edition section 7.2)
    - Computationally cheaper than:
      - leave k-out cross validation
      - k-fold cross validation

11



"The Elements of Statistical Learning"  
2nd edition section 7.2 page 222.  
[https://en.wikipedia.org/wiki/Cross-validation\\_\(statistics\)](https://en.wikipedia.org/wiki/Cross-validation_(statistics))

12

Let's review these terms

12





## Statistical Adjustments

• Attempt to estimate the value of a statistic or the performance of a model on new data using only training data.

• Examples:

• Sample size adjustment for variance: writing  $\sum_{i=1}^n (x_i - \bar{\mu})^2 / (n - 1)$  instead of  $\sum_{i=1}^n (x_i - \bar{\mu})^2 / n$

• “adjusted R-squared”, in-sample p-values, “AIC”, “BIC”, ...

• Pointless to adjust in training sample quantities when you have enough data to try and estimate out of sample quantities directly (cross validation methods, train/test methods, or bootstrap methods).

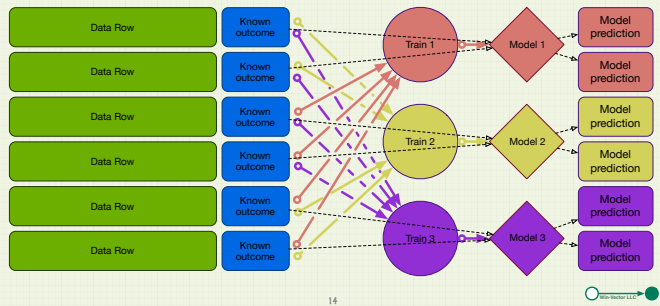


13

13

Does not matter when n is large. Can actually be quite complicated and require a lot of background to apply correctly. Prefer tools like the PRESS statistics to adjusted R-squared. Can use training mean against out of sample instances and so on.

## Leave k-out, k-way, and k-fold scoring



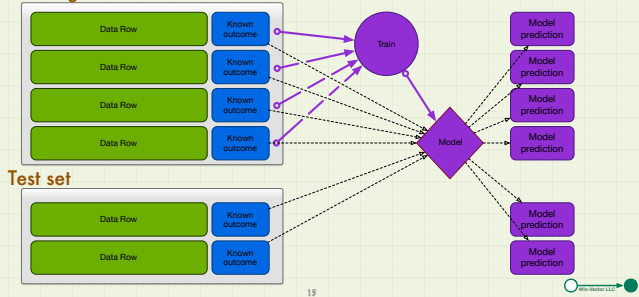
14

14

k-X cross validation methods are a procedural alternative. Shown: 3-fold cross validation. We try to simulate the performance of a model on new data by never applying a model to any data used to construct it. Which cross validation scheme you are using determines pattern of arrows. Common to all schemes: there are many throw away models. The larger the models the more like training on all of the available data they behave.

## Train/Test split

Training set



15

15

Test/Train split is an easier alternative that is less statistically efficient and depends on having good tools (that their selves cross-validate) during the training phase. Test set is held secret during model construction, tuning, and even early evaluation. Scoring in Train subset may in fact itself use both cross-validation and train/test subsplit methods. Actual model produced is scored on test set (though some data scientists re-train on the entire data set as a final “model polish” step).

## Train/Test split continued

- Statistically inefficient
  - Blocking issue for small data sets
  - Largely irrelevant for large data sets
- Considered “cross validation done wrong” by some statisticians
  - Cross validation techniques in fact estimate the quality of the *fitting procedure*, not the quality of the final returned model.
  - Test or held-out procedures directly estimate the performance of the actual model built.
  - Doesn't imitate the full structure of repeatedly drawing from a sampling distribution.

16



16

Splitting your available data into train and test is a way to try and *simulate* the arrival of future data. Like any simulation- it may fail. Controlled experiments are prospective designs that are somewhat more expensive and somewhat more powerful than this.

## Back to data science

17



17

Data science is a bit looser than traditional statistical practice and moves a bit faster; what does that look like?

## Data scientists rush in where statisticians fear to tread



- Large data sets
- Wide data sets
- Heterogeneous variables
- Colinear variables
- Noisy dependent variables
- Noisy independent variables

18



18

## We have to admit: data scientists are a flourishing species

- Must be something to be learned from that.
- What axioms (true or false) would be needed to explain their success?



19



19

## Data science axiom wish list

- Just a few:
  - Wish more data was always better.
  - Wish more variables were always better.
  - Wish you can retain some fraction of training performance on future model application.



20



20

Axioms that are true are true in the extreme.

## Is more data always better?



- In theory: yes (you could always simulate having less data by throwing some away).
- In practice: almost always yes.
- Absolutely for every algorithm every time? no.

21



21

## More data can be bad for a fixed procedure (artificial example)

- Statistics / machine learning algorithms that depend on re-sampling to supply *diversity* can degrade in the presence of extra data.
- Case in point: random forest over shallow trees can lose tree diversity (especially when there are duplicate or near-duplicate variables).

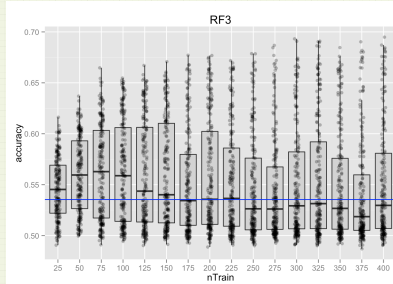
22



22

## Random forest example

- A data set where a random forest over shallow trees shows lower median accuracy on test data as we increase training data set size.
- (synthetic data set designed to hurt random forest, logistic model passes 0.85 accuracy)
- All code/data: <http://winvector.github.io/DS/>



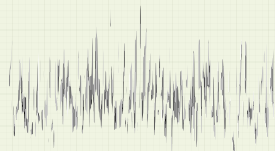
23

23

Random forest is a dominant machine learning algorithm in practice. This is a problem where logistic regression gets 85% accuracy as  $n$  increases, and the concept is reachable by the random forest model.

## Are more variables always better?

- In theory: yes.
  - Consequence of the non-negativity of mutual information.
  - Only true for training set performance, not performance on future instances.
- In practice: often.
- In fact: ridiculously easy to break:
  - Noise variables
  - Collinear variables
  - Near constant variables
  - Overfit



24



24

Note: collinear variables while damaging to prediction are nowhere near as large a hazard to prediction as they are to coefficient inference. And classic “x alone” methods of dealing with them become problematic in so called “wide data” situations.

## To benefit from more variables

- Need at least a few of the following:
  - Enough additional data to falsify additional columns.
  - Regularization terms / useful inductive biases.
  - Variance reduction / bagging schemes.
- Dependent variable aware pre-processing (variable selection, partial least squares, word2vec, and not principal components projection).

25



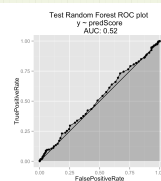
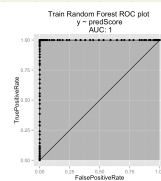
25

Principal components is a “independent variable only” or “x alone” transform, a good idea over curated homogeneous variable- not good over wild wide datasets. word2vec ( <https://code.google.com/p/word2vec/> ) can be considered not “x alone” as it presumably retains concept clusters from the grouping of data its training source (typically GoogleNews or Freebase naming); to it has an “y” (just not your “y”).

## Can't we keep at least some of our training performance?

- Common situation:
  - Near perfect fit on training data.
  - Model performs like random guessing on new instances.
  - Extreme over fit.
- One often hopes some regularized, ensemble, or transformed version of such a model would have at least some use on new instances.

26



26

## Not the case

- For at least the following common popular machine learning algorithms we can design a simple data set where we get arbitrarily high accuracy on training even when the dependent variable is generated completely independently of all of the independent variables.

- Decision Trees
- Logistic Regression
- Elastic Net Logistic Regression
- Gradient Boosting
- Naive Bayes
- Random Forest
- Support Vector Machine

(All code/data: <http://winvector.github.io/DS/> )

27

27

I.e. we see an arbitrarily good model on training, even when the model is possible.

Also have sometimes seen a reversal: the model is significantly worse than random on the test set. Being worse than random is likely a minor distribution change from training to test. The observed statistical significance is likely due to some process causing dependence between rows in a limited window (like serial correlation or bad sessionizing) and

## How did we design the counter examples?

• A lot of common machine learning algorithms fail in the presence of:

- Noise variables
- Duplicate examples
- Serial correlation
- Incompatible scales

• Punchline: all these things are common in typical under-curved real world data!



28

Some of these problems even break test/train exchangeability, one of the major justifications of machine learning.

## The analyst themselves can be a source of additional exotic “can never happen” biases

- Neighboring duplicate and near-duplicate rows (bad join/sessionizing logic).
- Features with activation patterns depending on the size of the training set (opportunistic feature generation/selection).
- Leakage of facts about evaluation set through repeated scoring (see “wacky boosting” by Moritz Hardt, which gives a reliable procedure to place high on Kaggle leaderboards without even looking at the data).



29

<http://blog.mrtz.org/2015/03/09/competition.html>

## What to do?



30

## Look for exploitable invariants to speed up machine learning process

### • Examples:

- Tree based methods are blind to any single monotone 1-1 input variable transforms
  - So don't need to try them
  - To some extent includes decision trees, random forests, rule ensembles, and gradient boosting
- AUC score is blind to any monotone 1-1 transform

31



31

## Look for universal methods

### • Example:

- Wald's complete class theorem
- Any admissible (minimum loss for all values of the unknown quantity to be inferred) inference procedure is Bayesian inference with an appropriate prior.

32



32

## Conclusions

- Data scientists, statisticians, and domain experts all see things differently.
  - Data science emphasizes procedures that are conceptually easy and become more correct when scaled to large data. Procedures can seem overly ambitious and as pandering to domain/business needs.
  - Statistics emphasizes procedures that are correct at all data scales, including difficult small data problems. Procedures can seem overly doctrinal and as insensitive to original domain/business needs.
  - Domain experts/scientists value correctness and foundation, over implementability.
- An effective data science team must work agilely, *understand* statistics, and develop *domain empathy*.
- We need a deeper science of structured back-testing.

33



33

It is equally arrogant to completely ignore domain science as it is to believe you can always quickly become a domain expert.

<http://www.win-vector.com/blog/2014/05/a-bit-of-the-agenda-of-practical-data-science-with-r/>

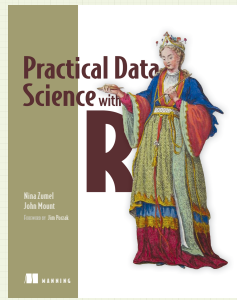
Better structured back-testing: i.e. invent procedures that obey appropriately adjusted “axioms of data science.”

# Thank you

34

For more, please check out my book,  
or contact me at [win-vector.com](http://win-vector.com)

Also follow our group on our blog  
<http://www.win-vector.com/blog/> or  
on Twitter @WinVectorLLC



34

