A Data-Driven Statistical Model for Predicting the Critical Temperature of a Superconductor

APS Meeting, March 9, 2018

Kam Hamidieh
Statistics and Data Sciences Department
UT Austin

Find All at:

https://github.com/khamidieh/predict_tc

Acknowledgement

Many thanks to Dr. Allan Macdonald, Sid W. Richardson Foundation Regents Chair in Physics, University of Texas at Austin.

Agenda

- Goal
- Data source discussion
- Data clean up
- Some model discussion
- (Live!) Demo

Goal

- Create a statistical model to predict T_c from elemental properties.
- Statistical model = Machine learning model

Data

- Data source: Superconducting Material Database maintained by Japan's National Institute for Materials Science (NIMS)
- Site: http://supercon.nims.go.jp/index_en.html
- Largest database of superconductors of all types
- Has errors
- Easily accessible

Data Clean Up

- Data accessed on July 24, 2017
- Started with 31,611 superconductors
- A combination of manual and software clean up left 21,263 superconductors.
- Took a bit of time.

Feature Extraction

Example: $Nb_{0.8}Pd_{0.2}$ with T_c = 1.98 K

Use the thermal conductivities of niobium and palladium to define a new “feature”:

Mean thermal conductivity = (54 + 71)/2 = 62.5 W/(m×K)

10

Feature & Description
Mean
Weighted mean
Geometric mean
Weighted geometric mean
Entropy
Weighted entropy
Range
Weighted range
Standard deviation
Weighted standard deviation

8

Variable
Atomic Mass
First Ionization Energy
Atomic Radius
Density
Electron Affinity
Fusion Heat
Thermal Conductivity
Valence

More on Feature Extraction:

- I had done some preliminary analysis to find out which elemental properties may be useful.

- A lot of it was driven by using properties with no missing values.

- No MAGPIE. No Aflow. (Didn't know about them.)
  - Good: Gain intuition
  - Bad: lots of coding

Final Data

- Data size = 21,263 rows by 82 Columns

- 82 = 81 features extracted for each superconductor (10 features × 8 element properties plus 1 features for the total number of elements) + T_c

- Features are highly correlated.

Main Model

- XGBoost = eXtreme Gradient Boost
- Set up: add a tree to an ensemble of trees in a sequential manner to improve the fit:

$$\text{Objective with respect to } f_t = \sum_{i=1}^{n} L(\underbrace{y_i}_{observed}, \underbrace{\hat{y}_i^{(t-1)} + f_t(x_i)}_{predicted}) + \Omega(f_t)$$

New Tree

Loss Function

Clever Penalty

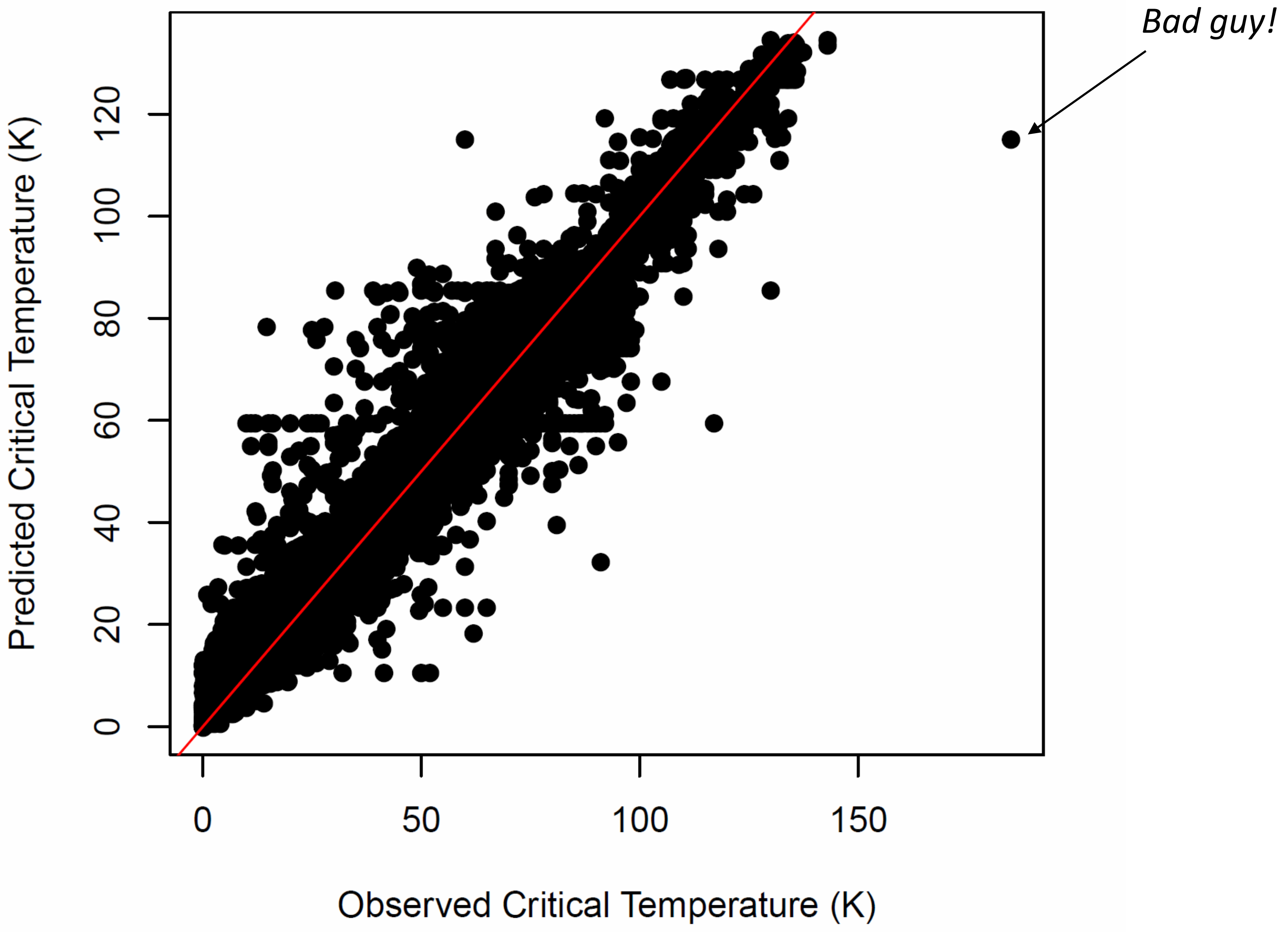
Anthony Goldbloom, CEO of Kaggle (now owned by Google):

"It used to be random forest that was the big winner, but over the last six months a new algorithm called XGBoost has cropped up, and it's winning practically every competition in the structured data category." (2015?)

Model Summary

- The model is tuned over various grid.

- Based on cross validation:
  - Out of sample root-mean-squared-error ≈ 9.5 K
  - Out of sample $R^2 \approx 0.92$



Bad guy!
Predicted Critical Temperature (K)
Observed Critical Temperature (K)
0
20
40
60
80
100
120
0
50
100
150

Demo:

https://github.com/khamidieh/predict_tc

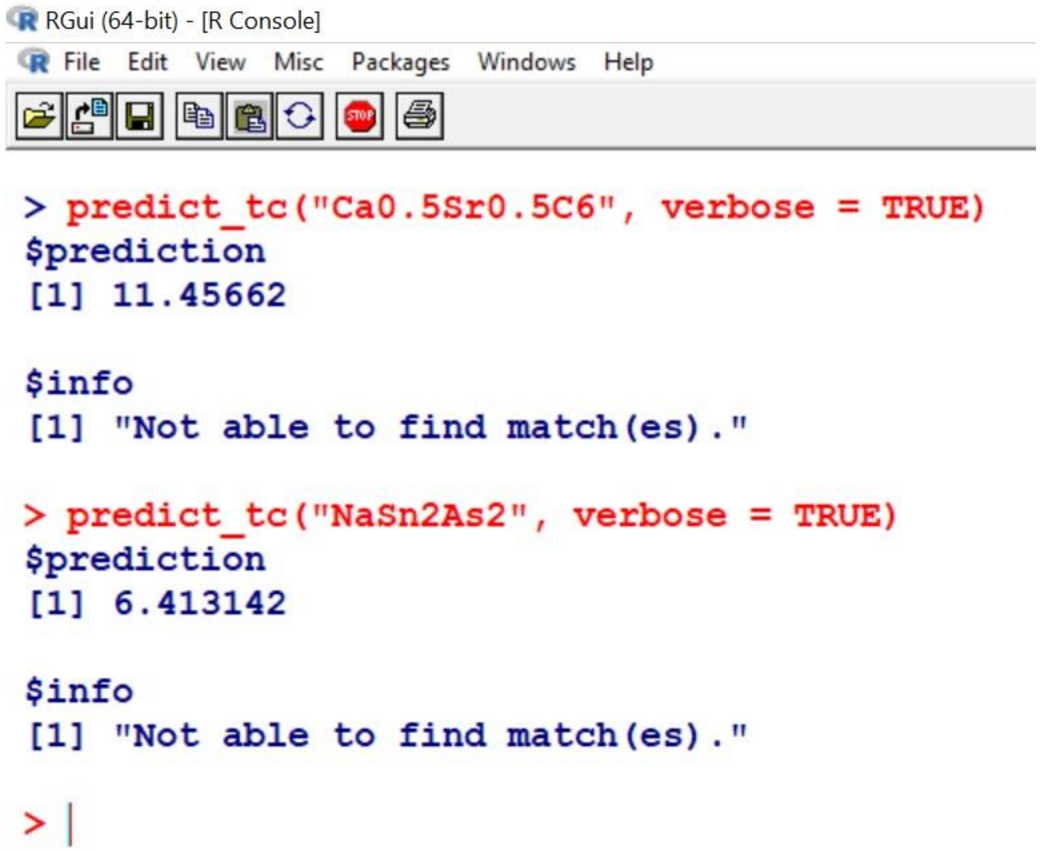
RGui (64-bit) - [R Console]

File Edit View Misc Packages Windows Help

```
> predict_tc("Ba0.2La1.8Cu1O4", verbose = TRUE)
$prediction
[1] 24.44241

$info
      critical_temp          material
1             29.00   Ba0.2La1.8Cu1O4
934           28.00   La1.8Ba0.2Cu1O4
1053          31.00   La1.8Ba0.2Cu1O4
1277          25.60   La1.8Ba0.2Cu1O4
1798          21.00   La1.8Ba0.2Cu1O4
2272          23.50 La1.801Ba0.199Cu1O4
2342          20.90   La1.8Ba0.2Cu1O4
2907          32.50   La1.8Ba0.2Cu1O4
3533          16.50   La1.8Ba0.2Cu1O4
7041          17.90   La1.8Ba0.2Cu1O4
9684          38.00   La1.8Ba0.2Cu1O4
20338          9.38   La1.8Ba0.2Cu1O4
20653         23.40   La1.8Ba0.2Cu1O4

> predict_tc("MgB2")
[1] 35.50066
> predict_tc("Hg")
[1] 4.076086
```


RGui (64-bit) - [R Console]
File Edit View Misc Packages Windows Help
STOP
> predict_tc("Ca0.5Sr0.5C6", verbose = TRUE)
$prediction
[1] 11.45662
$info
[1] "Not able to find match(es)."
> predict_tc("NaSn2As2", verbose = TRUE)
$prediction
[1] 6.413142
$info
[1] "Not able to find match(es)."
>

RGui (64-bit) - [R Console]

File Edit View Misc Packages Windows Help

```
> predict_tc("H2S", verbose = T)
$prediction
[1] 115.1063

$info
      critical_temp material
24024            60     H2S1
24025           185     H2S1

> predict_tc("FCl", verbose = T)
$prediction
[1] 31.7408

$info
[1] "Not able to find match(es)."

> predict_tc("mgB2", verbose = T)
Error in count.elements(formula) :
  'mgB2' is not a simple chemical formula
> |
```

Comments

- Features extracted based on thermal conductivity, atomic radius, valence, electron affinity, and atomic mass contribute the most to the model's predictive accuracy.
- Root-mean-squared-error is too global.
- The model probably won't predict out of the range of the observed T_c.
- Many potential variables are missing: pressure, crystal structure, manufacturing method, etc.
- Excellent project idea: predict if a material is a superconductor or not. However, we would need a ***large*** database of non-superconductors.

Thank you!

Questions or comments?