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

Kam Hamidieh
khamidieh@gmail.com
Statistics and Data Sciences Department
The University of Texas at Austin

March 3, 2018

Abstract

We estimate a statistical model to predict the superconducting critical temperature based on the features extracted from the superconductor's chemical formula. The statistical model gives reasonable out-of-sample predictions: ±9.5 K based on root-mean-squared-error. Features extracted based on thermal conductivity, atomic radius, valence, electron affinity, and atomic mass contribute the most to the model's predictive accuracy. It is crucial to note that our model does not predict whether a material is a superconductor or not; it only gives predictions for superconductors.

1 Introduction

We derive "features" (or predictors) based on the properties of elements in the superconductor that could be helpful in predicting the supeconducting critical temperature (T_c). For example, consider $\mathrm{Nb}_{0.8}\mathrm{Pd}_{0.2}$ with $T_c = 1.98$ K. We can derive a feature based on the average thermal conductivities of the elements. Niobium and palladium's thermal conductivity coefficients are 54 and 71 W/(m×K) respectively. The mean thermal conductivity is $(54 + 71)/2 = 62.5$ W/(m×K). We can treat the mean thermal conductivity variable as a feature to predict T_c. In total, we define and extract 81 features from each superconductor.

The software to predict T_c and the data are available at `https://github.com/khamidieh/predict_tc`.

The superconductor data comes from the Superconducting Material Database maintained by Japan's National Institute for Materials Science (NIMS) at `http://supercon.nims.go.jp/index_en.html`. After some data preprocessing, 21,263 superconductors are used.

We tried various statistical models but we eventually settled on two: A multiple regression model which serves as a baseline model, and a gradient boosted model as the main prediction model.

Gradient boosted models create an ensemble of trees to predict a response. The trees are added in a sequential manner to improve the model by accounting for the points which are difficult to predict. Once a gradient boosted model is fitted, the weighted average of all the trees is used to give a final prediction. Gradient boosted models predict well because they are able to account for the complex interactions and correlations among the features.

The boosted models were first introduced by Freund (1995). They were generalized to *gradient* boosting by Friedman (2001). We use the latest improvement called XGBoost (eXtreme Gradient Boosting) by Chen and Guestrin (2016), and the associated open-source R implementation of XGBoost by Chen *et al.* (2018a). XGBoost is also available in other popular programming languages such as python and Julia. The full source code is at `https://github.com/dmlc/xgboost`.

Anthony Goldbloom, CEO of Kaggle[1], the premier data competition site, stated: “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.” You can see the talk at `https://www.youtube.com/watch?v=GTs5ZQ6XwUM`. Outside the competition realm, XGBoost has been successfully applied in disease prediction by Chen *et al.* (2018b), and in quantitative structure activity relationships studies by Sheridan *et al.* (2016).

Our XGBoost model gives good predictions: an out-of-sample error of about 9.5 K based on root-mean-squared-error (rmse), and an out-of-sample R^2 values of about 0.92. The numbers for the multiple regression model are about 17.6 K and 0.74 for the out-of-sample rmse and R^2 respectively. The multiple regression is simply used to establish a baseline model.

We are able to assess the importance of the features in prediction accuracy. Features defined based on thermal conductivity, atomic radius, valence, electron affinity, and atomic mass are the most important features in predicting T_c. On the downside, simple conclusions such as the exact nature of the relationship between the features and T_c can’t be inferred from the XGBoost model.

Valentin *et al.* (2017) also create a model to predict T_c. Our approach is different than Valentin *et al.* (2017) in the following ways: (1) We use XGBoost versus random forests, (2) we use a larger data set and include superconductors with carbon (around 21,263 versus 12,000), (3) we use a single large model to obtain predictions rather than a cascade of models, (4) we create a larger number features *only* from the element properties, and (5) most importantly, we quantify the out-of-sample prediction error.

2 Data Preparation

This section describes the detailed steps for data preparation and feature extraction. Subsection (2.1) describes how the element data is obtained and processed. Subsection (2.2) describes the data preparation from NIMS Superconducting Material Database. Subsection (2.3) details how the features are extracted.

[1] Google purchased Kaggle in 2017.

Variable	**Units**	**Description**
Atomic Mass	atomic mass units (AMU)	average of the proton and neutron rest masses
First Ionization Energy	kilo-Joules per mole (kJ/mol)	energy required to remove a valence electron
Atomic Radius	picometer (pm)	distance from the nucleus to the outermost electron
Density	kilograms per meters cubed (kg/m^3)	density at standard temperature and pressure
Electron Affinity	kilo-Joules per mole (kJ/mol)	energy required to add an electron to a neutral atom
Fusion Heat	kilo-Joules per mole (kJ/mol)	energy to change from solid to liquid without temperature change
Thermal Conductivity	watts per meter-Kelvin (W/(m × K))	thermal conductivity coefficient κ
Valence	no units	typical number of chemical bonds formed by the element

Table 1: This table shows the properties of an element which are used for creating features to predict T_c.

2.1 Element Data Preparation

The element data with 46 variables are obtained by using the `ElementData` function from Mathematica Version 11.1 by Wolfram and Research (2017). Appendix (A) lists the information sources for the element properties used by `ElementData`. The first ionization energy data came from `http://www.ptable.com/` and is merged with the Mathematica data. About 12% of the entries are missing.

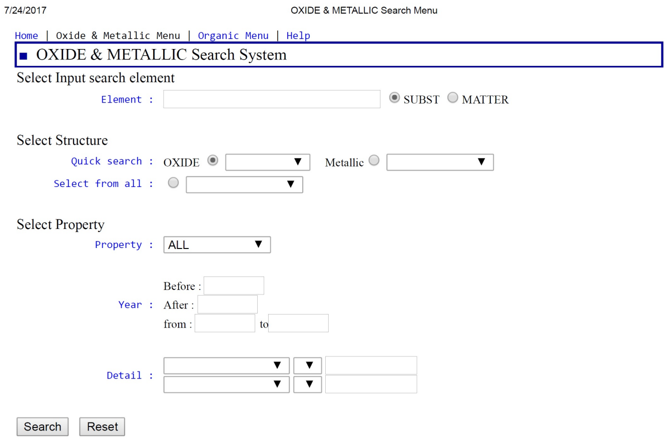
In choosing the properties, we are guided by Conder (2016) but we also use our judgement to pick certain properties. For example, we drop the boiling point variable, and instead use the fusion heat variable which has all nonnegative values (this makes taking logs easier), no missing values, and highly correlated with the boiling point variable. We had also gained some experience and insight creating some initial models for predicting T_c of elements only. We settle on 8 properties shown in table (1).

The atomic radii of La and Ce are missing so we replace them with their covalent radii since atomic radii and covalent radii have very high correlation. We add a small constant of 1.5 to the electron affinity values of all the elements to prevent issues when taking logarithm of 0.

2.2 Superconducting Material Data Preparation

Superconducting Material Database is supported by the NIMS, a public institution based in Japan. The database contains a large list of superconductors, their critical temperatures, and the source references mostly from journal articles. To our knowledge, this is the most comprehensive database of superconductors. Access to the database requires a login id and password but this is provided with a simple registration process.

We accessed the data on July 24, 2017 at `http://supercon.nims.go.jp/supercon/material_menu`. Once logged in, we chose "OXIDE & METALLIC" material. Figure (1) shows a screen shot of the menu. We clicked on the "search" button to get *all* the data. We obtained 31,611 rows of data in a comma separated file format. The key columns (variables) were "element", the chemical formula of the material, and "Tc", the critical temperature. Variable "num" was a unique identifier for each row. Column "refno" contained links to the referenced source. The next few steps describe the manual clean up process:



7/24/2017
OXIDE & METALLIC Search Menu
Home | Oxide & Metallic Menu | Organic Menu | Help
■ OXIDE & METALLIC Search System
Search
Reset

Select Input search element

Element : ● SUBST ○ MATTER

Select Structure

Quick search : OXIDE ● Metallic ○

Select from all : ○

Select Property

Property : ALL

Year : Before : After : from : to

Detail :

Figure 1: This is a screen shot of from Superconducting Material Database accessed on July 24, 2017.

1. We remove columns “ma1” to “mj2”.

2. We sort the data by “Tc” from the highest to lowest.

3. The critical temperature for the following “num” variables are mistakenly shifted by one column to the right. We fix these by recording them under the “Tc” column: 31020, 31021, 31022, 31023, 31024, 31025, 153150, 153149, 42170, 42171, 30716, 30717, 30718, 30719,150001, 150002, 150003, 150004, 150005, 150006, 150007, 30712, 30713, 30714, 30715.

4. The following are removed since the critical temperatures seemed to have been misrecorded; They have critical temperatures over 203 K which as of July 2017 was the highest reliable recorded critical temperature. La0.23Th0.77Pb3 (num = 111620), Pb2C1Ag2O6 (num = 9632), Er1Ba2Cu3O7-X (num = 140)

5. All rows with “Tc” = 0 or missing are removed.

6. Columns with headings “nums”, “mo1”, “mo2”, “oz”, “str3”, “tcn”, “tcfig”, “refno” are removed.

7. We manually chang all materials with oxygen content formula such as O7-X to the best oxygen content approximation. For example, O7-X is changed to O7, O5+X is changed to O5, etc. This certainly introduces some error into our data but it is impossible to go document by document to get better estimates of the oxygen contents. At this point our data has two columns: “element” and “Tc”.

8. We use R statistical software by R Core Team (2017) and the CHNOSZ package by Dick (2008) to perform a preliminary check of the validity of the chemical formulas. The CHNOSZ package has a function `makeup` which reads the chemical formula in string format and breaks up the formula into the elements and their ratios. In some cases, it throws an error or a

warning when the chemical formula does not make sense. For example it throws a warning message if Pb-2O is checked; Negative number of Pb does not make sense. However, the function does not check whether the material could actually exist. See figure (2) to get a sense of how this function works. With the help of the CHNOSZ package, we make the following modifications:

(a) Yo975Yb0.025Ba2Cu3O, Yo975Yb0.025Ba2Cu3O, Yo975Yb0.025Ba2Cu3O are removed. There is no element with the symbol Yo. It's likely that Y0.975 was misrecorded as Yo975 but we can't be sure.

(b) Bi1.7Pb0.3Sr2Ca1Cu2O0, La1.85Nd0Ca1.15Cu2O5.99, Bi0Mo0.33Cu2.67Sr2Y1O7.41, Y0.5Yb0.5Ba2Sr0Cu3O7 are removed since some elements had coefficients of zero.

(c) Y2C2Br0.5!1.5 is removed. The exclamation sign throws an error message.

(d) Y1Ba2Cu3O6050 is removed. The coefficient of 6050 for oxygen is possibly a mistake.

(e) Hg1234O10 is removed. The coefficient of 1234 for mercury is possibly a mistake.

(f) Nd185Ce0.15Cu1O4 is removed. The coefficient of 185 for Neodymium is possibly a mistake. There is a Nd1.85Ce0.15Cu1O4 already in the data.

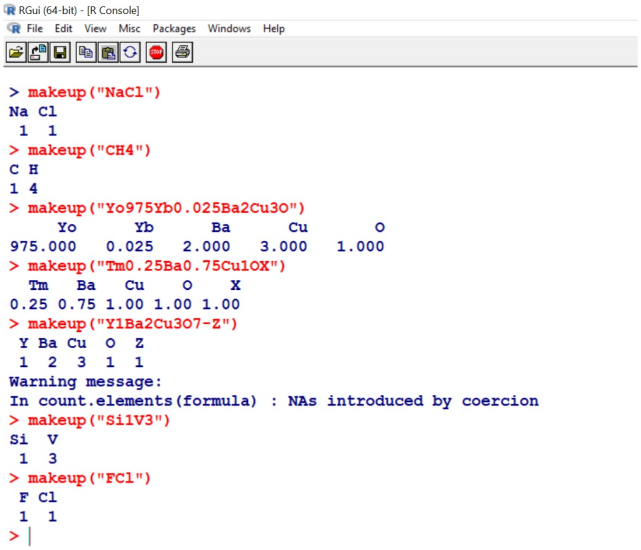
(g) Bi1.6Pb0.4Sr2Cu3Ca2O1013 is changed to Bi1.6Pb0.4Sr2Cu3Ca2O10.13 since nearby rows in the data have formulas with O10.xx.

(h) Y1Ba2Cu285Ni0.15O7 is changed to Y1Ba2Cu2.85Ni0.15O7 since nearby rows in the data have formulas with Cu2.xx.

9. The column headings of "Tc" and "element" are changed to "critical_temp" and "material" respectively.

6750 rows are left out because T_c is either zero or missing. At this point we have 24,861 rows of data.

The rest of the data preparation is done in R Core Team (2017). We exclude any superconductor that has an element with an atomic number greater than 86. This eliminates an additional 973 rows of data. For example, superconductors that have uranium are left out. We remove the repeating rows. It would be impossible to manually check to see whether the repeated rows are genuine independent reports from independent experiments or they are just duplicate reportings. After all the data preparation and clean up, we end up with 21,263 rows of data or about 67% of the original data we started with.


RGui (64-bit) - [R Console]
File Edit View Misc Packages Windows Help

```
> makeup("NaCl")
Na Cl
 1  1
> makeup("CH4")
C H
1 4
> makeup("Yo975Yb0.025Ba2Cu3O")
     Yo      Yb      Ba      Cu       O
975.000   0.025   2.000   3.000   1.000
> makeup("Tm0.25Ba0.75Cu1OX")
  Tm    Ba   Cu    O    X
0.25 0.75 1.00 1.00 1.00
> makeup("Y1Ba2Cu3O7-Z")
 Y Ba Cu  O  Z
 1  2  3  1  1
Warning message:
In count.elements(formula) : NAs introduced by coercion
> makeup("Si1V3")
Si  V
 1  3
> makeup("FCl")
 F Cl
 1  1
> |
```

Figure 2: This screen shot is intended give you a sense of how the CHNOSZ package by Dick (2008) works. The first two materials NaCl and CH4 are correctly broken up. (These two are not superconductors and they are shown for illustration purposes.). $Yo_{975}Yb_{0.025}Ba_2Cu_3O$ was a material in the database but this is obviously a mistake since no element with the symbol Yo exists. The same is true for the next material with X. However, no warnings are issued. A warning is issued for $Y_1Ba_2Cu_3O_{7-Z}$. The next material SiV_3 was in the database and is correctly broken up. FCl is just given as another example. It is not a superconductor and was not in the database. The `makeup` command correctly breaks up the material but obviously does not check for the existence of FCl.

Feature & Description	**Formula**	**Sample Value**
Mean	$= \mu = (t_1 + t_2)/2$	35.5
Weighted mean	$= \nu = (p_1 t_1) + (p_2 t_2)$	44.43
Geometric mean	$= (t_1 t_2)^{1/2}$	33.23
Weighted geometric mean	$= (t_1)^{p_1} (t_2)^{p_2}$	43.21
Entropy	$= -w_1 \ln(w_1) - w_2 \ln(w_2)$	0.63
Weighted entropy	$= -A \ln(A) - B \ln(B)$	0.26
Range	$= t_1 - t_2 \ (t_1 > t_2)$	25
Weighted range	$= p_1 t_1 - p_2 t_2$	37.86
Standard deviation	$= [(1/2)((t_1 - \mu)^2 + (t_2 - \mu)^2)]^{1/2}$	12.5
Weighted standard deviation	$= [p_1 (t_1 - \nu)^2 + p_2 (t_2 - \nu)^2)]^{1/2}$	8.75

Table 2: This table summarizes the procedure for feature extraction from material's chemical formula. The last column serves as an example; features based on thermal conductivities for $Re_7Zr_1$are derived and reported to two decimal places. Rhenium and Zirconium's thermal conductivity coefficients are $t_1 = 48$ and $t_2 = 23$ W/(m×K) respectively. Here: $p_1 = \frac{6}{7}, p_2 = \frac{1}{7}, w_1 = \frac{48}{71}, w_2 = \frac{23}{71}, A = \frac{p_1 w_1}{p_1 w_1 + p_2 w_2} \approx 0.926, B = \frac{p_2 w_2}{p_1 w_1 + p_2 w_2} \approx 0.074$.

2.3 Feature Extraction

In this section, we describe the feature extraction process through a detailed example: Consider Re_7Zr_1 with $T_c = 6.7$ K, and focus on the features extracted based on thermal conductivity.

Rhenium and Zirconium's thermal conductivity coefficients are $t_1 = 48$ and $t_2 = 23$ W/(m×K) respectively. The ratios of the elements in the material are used to define features:

$$p_1 = \frac{6}{6+1} = \frac{6}{7}, \quad p_2 = \frac{1}{6+1} = \frac{1}{7}. \tag{1}$$

The fractions of total thermal conductivities are used as well:

$$w_1 = \frac{t_1}{t_1 + t_2} = \frac{48}{48 + 23} = \frac{48}{71}, \quad w_2 = \frac{t_2}{t_1 + t_2} = \frac{23}{48 + 23} = \frac{23}{71}. \tag{2}$$

We need a couple of intermediate values based on equations (1) and (2):

$$A = \frac{p_1 w_1}{p_1 w_1 + p_2 w_2} \approx 0.926, \quad B = \frac{p_2 w_2}{p_1 w_1 + p_2 w_2} \approx 0.074.$$

Once we have obtained the values p_1, p_2, w_1, w_2, A, and B, we can extract 10 features from Rhenium and Zirconium's thermal conductivities as shown in table (2).

We repeat the same process above with the 8 variables listed in table (1). For example, for features based on atomic mass, just replace t_1 and t_2 with the atomic masses of Rhenium and Zirconium respectively, then carry on with the calculations of p_1, p_2, w_1, w_2, A, B, and finally calculate the 10 features defined in table (2). This gives us $8 \times 10 = 80$ features. One additional features, a numeric variable counting the number of elements in the supercondutor, is also extracted. We end up with 81 features in total.

In summary: We have data with 21,263 rows and 82 columns: 81 columns corresponding to the features extracted and 1 column of the observed T_c values.

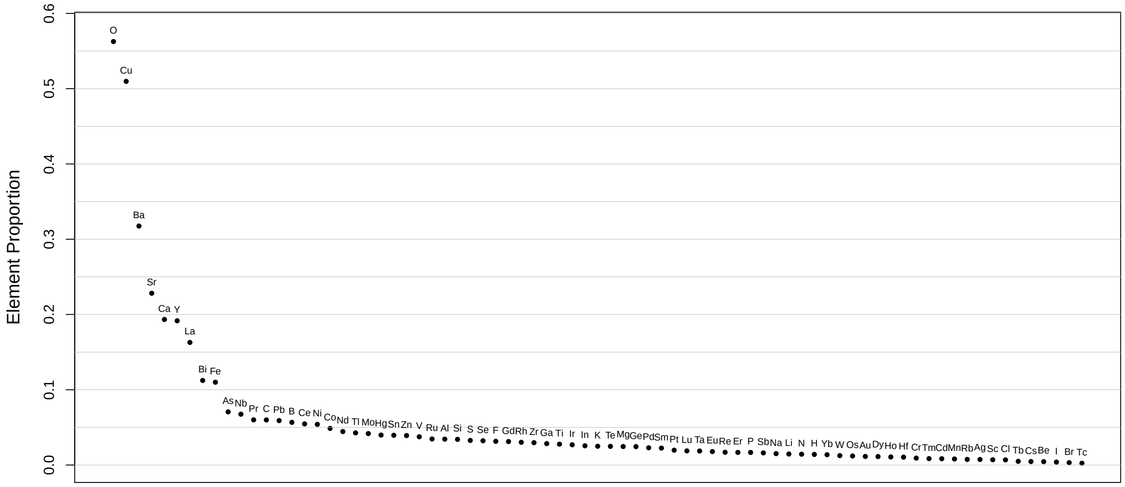


Figure 3: This figure shows the proportions of the superconductors that had each element.

We also considered but did not implement features that simply indicate whether an element is present in the superconductor or not. For example, we could have had a column that indicated whether say oxygen is in the material or not. However, this approach would have added a large number of indicator variables to our data, made model selection and assessment too complicated, and increased the chances of over-fitting.

3 Analysis

This section has two parts: Basic summaries of the data are given in subsection (3.1). The statistical models are described in subsection (3.2).

3.1 Descriptive Analysis

Figure (3) shows the proportions of the superconductors that had each element. For example, Oxygen is present in about 56% of the superconductors. Copper, barium, strontium, and calcium are the next most abundant elements.

Iron-based superconductors and cuprates are of particular interest in many research groups so we report some summary statistics in table (3). Iron is present in approximately 11% of the superconductors. The mean T_c of superconductors with iron is 26.9 ± 21.4 K. The non-iron containing superconductors' mean is 35.4 ± 35.4 K; the mean and standard deviations happened to be the same. A t-distribution based 95% confidence interval suggests that iron containing superconductors' mean T_c is lower than the non-iron's by 7.4 to 9.5 K. Cuprates comprise approximately 49.5% of the superconductors. The cuprates' mean T_c is 59.9 ± 31.2 K. The non-cuprates' mean T_c is 9.5 ± 10.7 K. A t-distribution based 95% confidence interval indicates that the cuprates' mean T_c

	Size	Min	Q1	Median	Q3	Max	Mean	SD
Iron	2339	0.02	11.3	21.7	35.5	130.0	26.9	21.4
Non-Iron	18924	0.0002	4.8	19.6	68.0	185.0	35.4	35.4
Cuprate	10532	0.001	31.0	63.1	86.0	143	59.9	31.2
Non-Cuprate	10731	0.0002	2.5	5.7	12.2	185	9.5	10.7

Table 3: This table reports summary statistics on iron-based versus non-iron, and cuprate versus non-cuprate superconductors. The Size is the total number of observations of the material out of 21,263 materials. For example, 2,339 out of 21,263 materials contained iron. The rest of the columns report summary statistics for the observed critical temperatures (K): min = minimum, Q1 = first quartile, Median = median, Q3 = third quartile, Max = maximum, and SD = standard deviation.

Min	Q1	Median	Q3	Max	Mean	SD
0.00021	5.4	20	63	185.0	34.4	34.2

Table 4: This table reports the summary statistics for the critical temperatures values (K) of all 21,263 superconductors. The column headers are the min = minimum , Q1 = first quartile , median , Q3 = third quartile, Max = maximum, and SD = standard deviation of the superconducting critical temperatures (K).

is higher than the non-cuprates' mean T_c by 49.8 to 51.0 K.

Figure (4) shows the histogram of T_c values. The values are right skewed with a bump around 80 K. Table (4) shows the summary statistics for T_c values.

Figure (5) shows the mean T_c grouped by elements. Mercury containing superconductors have the highest T_c at around 80 K on average. However, this is not the full story. Figure (6) shows the standard deviation of T_c grouped by elements. Although mercury containing superconductors have the highest T_c on average, these same materials show the fourth highest variability in T_c. In fact, a plot of the mean T_c versus the standard deviation of T_c in figure (7) shows that on average the higher the mean T_c, the higher the variability in T_c per element.

The average absolute value of the correlation among the features is 0.35. This indicates that the features are highly correlated. Motivated by this result, we attempted to reduce the dimensionality of the data using principal component analysis (PCA). However, our PCA analysis did not show any benefits in reducing the dimensionality since a large number of principal components were needed to capture a substantial percentage of the data variation; we abandoned the PCA approach.

3.2 Model Analysis

In this section we discuss the results of the multiple regression model, and the XGBoost model. We tried a few classical models including multiple regression with interactions, principal component regression, and partial least squares but none of these make any substantial improvements to the XGBoost model. We also tried random forests but they were too slow to tune given the data size

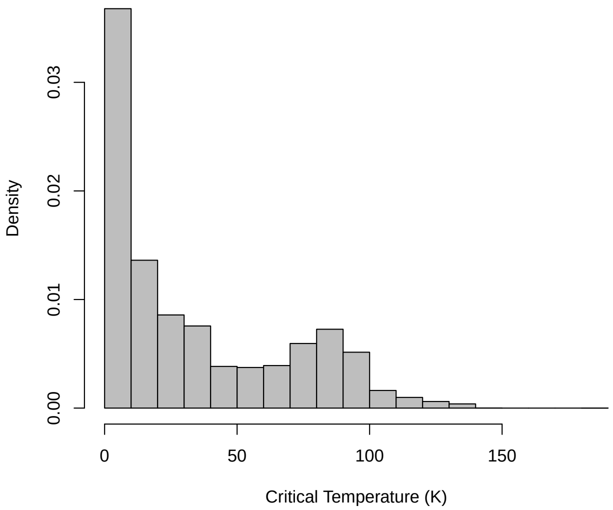


Figure 4: This figure shows the distribution of the superconducting critical temperatures (K) of all 21,263 superconductors.

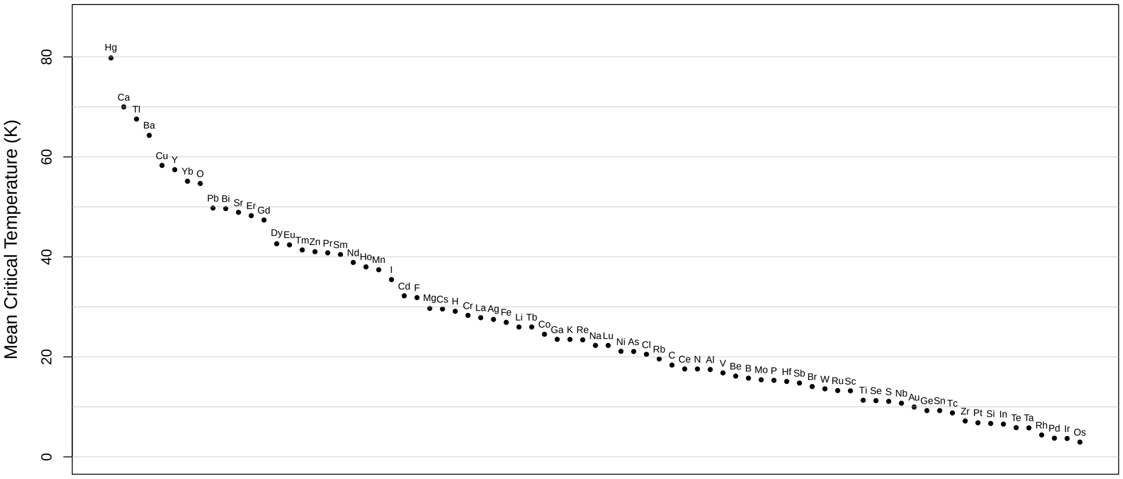


Figure 5: This figure shows the mean superconducting critical temperature grouped by elements. On average, mercury containing materials had the highest superconducting critical temperature followed by calcium and so on.

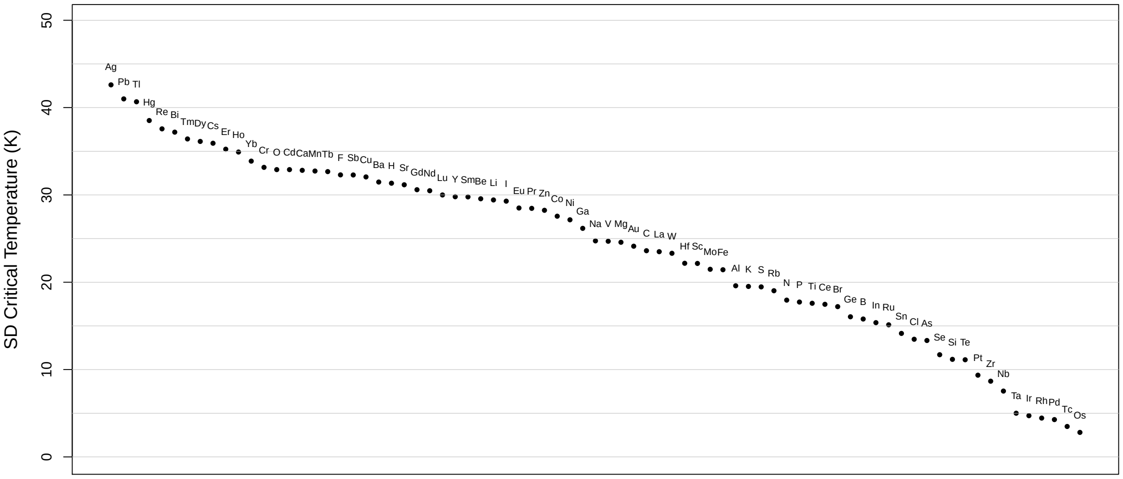


Figure 6: This figure shows the standard deviation (SD) of critical temperature grouped by elements. Silver containing materials had the highest variability followed by lead and so on.

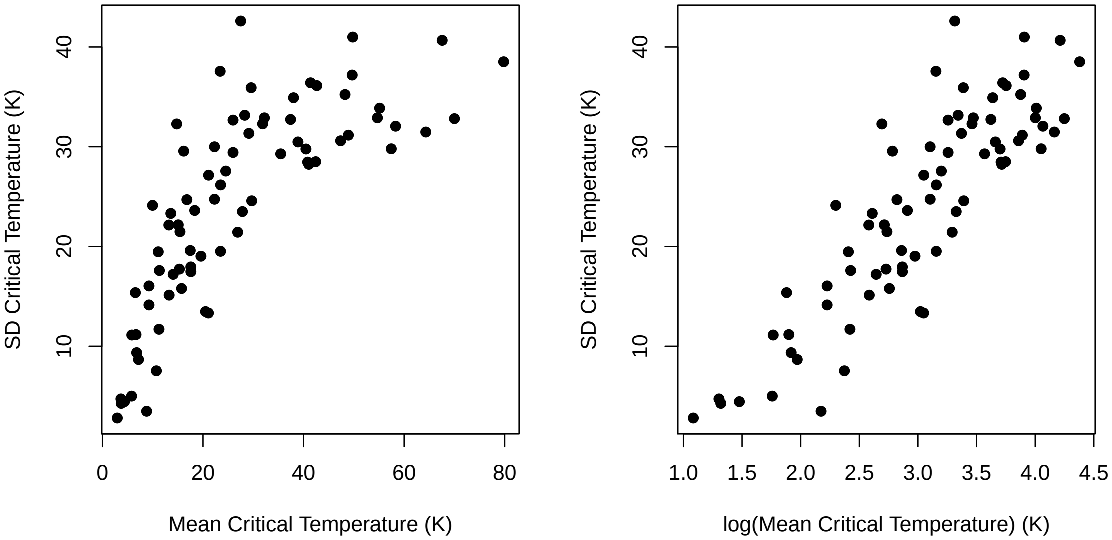


Figure 7: The left panel shows the relationship between the mean critical temperature and standard deviation (SD) per element. The right panel shows the logarithm of the mean critical temperature versus SD. On average the higher the mean critical temperature, the higher the variability in critical temperature per element.

and the number of features. Scalability and speed are important advantages of using XGBoost over random forests; See Chen and Guestrin (2016).

The prediction performance of the models are compared by using out-of-sample rmse. The out-of-sample rmse is estimated by the following cross validation procedure:

Out-Of-Sample RMSE Estimation Procedure:

1. At random, divide the data into 2/3 train data and 1/3 test data.
2. Fit the model using the train data.
3. Predict T_c of the test data.
4. Obtain an estimate of the out-of-sample mean-squared-error (mse) by using the predictions from the last step and the observed T_c values in the test data:

$$\text{out-of-sample mse} = \text{Average of (observed - predicted)}^2$$

5. Repeat steps 1 through 4, 25 times to collect 25 out-of-sample mse's.
6. Take the mean of the 25 collected out-of-sample mse's and report the square root of this average as the final estimate of the out-of-sample rmse.

3.2.1 The Multiple Regression Model

The multiple regression model's out-of-sample rmse estimated by the procedure above is about 17.6 K. The out-of-sample R^2 is about 0.74. Figure (8) shows the predicted T_c versus the observed T_c when we use all the data to fit the model. The line has an intercept of zero and a slope of 1. The plot indicates that the multiple regression model under-predicts T_c of high temperature superconductors since many predicted points are below the line for the high temperature superconductors. The model over-predicts low temperature superconductors' T_c. The multiple regression model simply serves as a baseline model and should not be used for prediction. There would be no use in predicting T_c using a sophisticated model such as XGBoost, if a simple multiple regression model does a good job. Here, the XGBoost model vastly improves the prediction accuracy.

3.2.2 The XGBoost Model

Before we go on, we give a brief description of XGBoost set up. XGBoost is described in detail in Chen and Guestrin (2016). A readable summary is given at `https://xgboost.readthedocs.io/en/latest/model.html`. Hastie *et al.* (2009) and Izenman (2008) give general overviews on boosting as well.

The functional form of XGBoost is:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i),$$

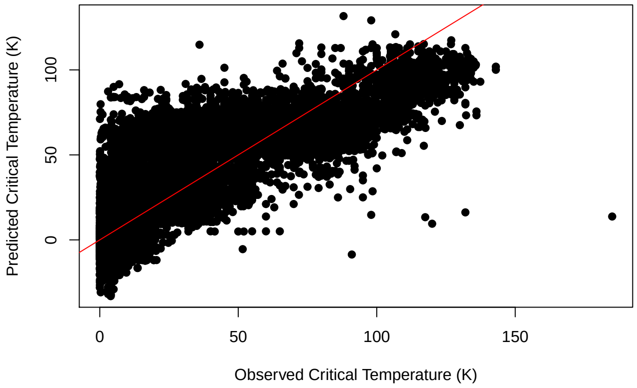


Figure 8: This plot shows the predicted superconducting critical temperatures (K) versus the observed superconducting critical temperatures (K) based on the multiple regression model. The out-of-sample rmse is about 17.6 K. The out-of-sample R^2 is about 0.74.

where x_i is the ith input feature vector, $\hat{y}_i$ is the predicted response, and $f_1, \ldots, f_K$ is a sequence of trees. The t-th tree f_t is added by minimizing the following objective function:

$$\text{Objective with respect to } f_t = \sum_{i=1}^{n} L\big(\underbrace{y_i}_{observed} , \underbrace{\hat{y}_i^{(t-1)} + f_t(x_i)}_{predicted} \big) + \Omega(f_t), \tag{3}$$

where L is the desired loss function, n is the total sample size, y_i's are the response values, $\hat{y}_i^{(t-1)}$ is the ith predicted responses at the $t-1$ step, and Ω is a penalty function. The form of Ω is:

$$\Omega(f) = \gamma T + (1/2)\lambda \sum_{j=1}^{T} w_j^2, \tag{4}$$

where T is the number of leaves in each tree, w_j's are the leaf weights, and λ and γ are regularization parameters. The goal here is to add a new tree f_t to the overall ensemble of trees to minimizes the loss between the observed and the predicted in equation (3), while preventing over-fitting by satisfying the penalty in equation (4). The addition of this penalty function to *each* tree in (4) is one major XGBoost differentiator from the established method by Friedman (2001). The penalty function appears to make a big difference in practice; see Chen and Guestrin (2016). Besides the clever penalty function, Chen and Guestrin (2016) implement numerous computational tricks to make their software scalable and very fast.

In addition to the penalty function, there are a number of tuning parameters that could reduce over-fitting and enhance the model's prediction performance; They are mainly: (1) column subsampling which means only a fraction of the features are chosen at random at each stage of adding a new tree, (2) a learning parameter $0 < \eta < 1$ which scales the contribution of each new tree, (3)

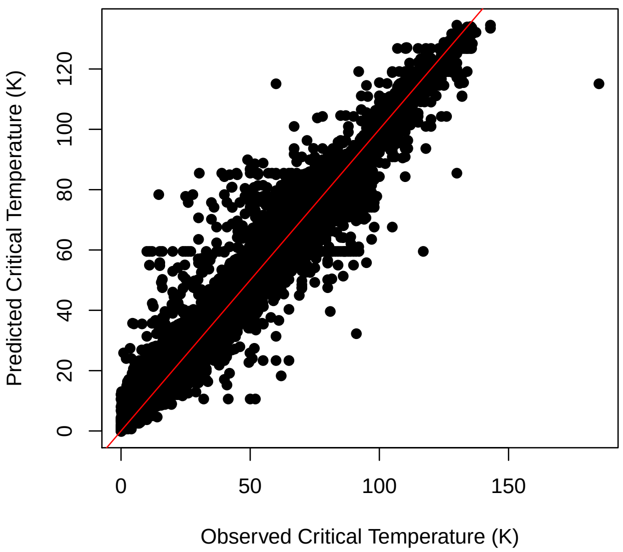


Figure 9: This plot shows the predicted critical temperatures versus observed critical temperatures (K) based on the XGBoost model. The out-of-sample rmse is 9.4 K. The out-of-sample R^2 is 0.92.

subsample ratio which means that XGBoost only uses a small percentage of the data to grow a new tree, (4) maximum depth of a tree, and (5) minimum child weight which is the minimum number of data points needed to be in each node.

To tune XGBoost, we first split the data at random to 2/3 train and 1/3 test data. Next, we create a grid - a grid contains all the possible combination of tuning parameters - with $\eta = 0.010, 0.015, 0.020$, column subsampling = 0.25, 0.5, 0.75, subsample ratio = 0.5, minimum node size = 1, 10, and maximum depth of a tree = 15, 16, ..., 24, 25. The total gird size is 198. This means that we need 198 different XGBoost models. For each model, 750 trees are grown. The rest of the XGBoost parameters are set to the default values. (This was not our only grid; we had done some experimentations with various grids before we decided to use this grid.) Finally, we evaluate the prediction accuracy of each model based on rmse at each tree = 1, 2, ..., 749, 750.

The best model (with the lowest out-of-sample rmse) turn out to be: $\eta = 0.02$, maximum depth = 16, minimum child weight = 1, column subsampling = 0.50, and a tree size of 374. To obtain the final out-of-sample rmse and R^2, we follow the 6 step procedure outlined at the begining of section (3.2). The procedure yield an out-of-sample rmse of 9.5 K, and a out-of-sample R^2 of 0.92. The out-of-sample rmse of 9.5 K has a very important interpretation: On average, the tuned XGBoost model will be off by about 9.5 K when predicting T_c.

Figure (9) shows the predicted T_c versus the observed T_c. No severe bias is discernable here but there are a number outliers visible.

Feature	**Gain**
range_ThermalConductivity	0.295
wtd_std_ThermalConductivity	0.084
range_atomic_radius	0.072
wtd_gmean_ThermalConductivity	0.047
std_ThermalConductivity	0.042
wtd_entropy_Valence	0.038
wtd_std_ElectronAffinity	0.036
wtd_entropy_atomic_mass	0.025
wtd_mean_Valence	0.022
wtd_gmean_ElectronAffinity	0.021
wtd_range_ElectronAffinity	0.016
wtd_mean_ThermalConductivity	0.015
wtd_gmean_Valence	0.014
std_atomic_mass	0.013
std_Density	0.010
wtd_entropy_ThermalConductivity	0.010
wtd_range_ThermalConductivity	0.010
wtd_mean_atomic_mass	0.009
wtd_std_atomic_mass	0.009
gmean_Density	0.009

Table 5: This figure shows the top 20 most important features based on the XGBoost gain. Here: wtd = weighted, gmean = geometric mean, std = standard deviation.

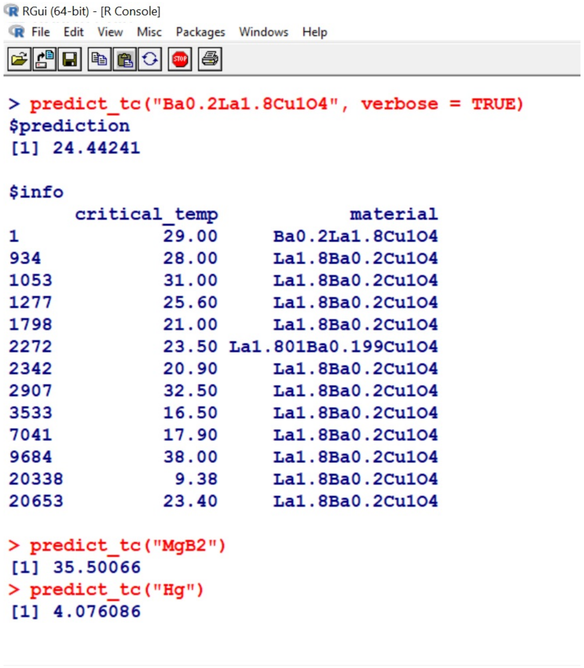
3.2.3 Feature Importance

Feature importance in XGBoost is measured by gain. The gain for a feature is defined as follows: Whenever a tree is split on a feature, the improvement in the objective function is recorded. The gain for the feature is then:

$$\text{The Gain for the Feature} = \frac{\text{Sum of the Gains for the Feature}}{\text{Sum of the Gains for All the Features}}.$$

Features with higher gain are more important.

Table (5) shows the top 20 most important features. Features extracted based on thermal conductivity, atomic radius, valence, electron affinity, and atomic mass appear to be the most important features. Also observe that features defined based on thermal conductivity, valence, electron affinity, and atomic mass appear most often on the list. This may suggest that these properties could be more important than other properties in predicting T_c.



```
> 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
```

Figure 10: This figure shows the software prediction results for $Ba_{0.2}La_{1.8}CuO_4$, MgB2, and Hg.

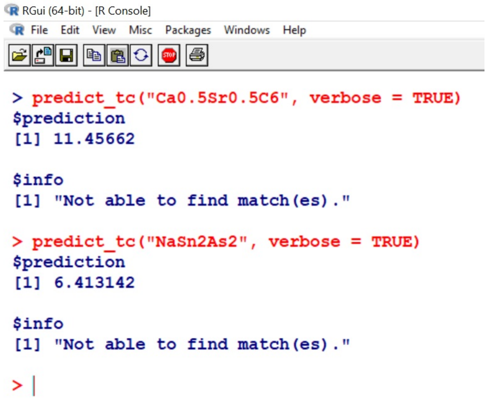
4 Prediction Software

We have put the code and software components prediction at `https://github.com/khamidieh/predict_tc`. The software is created using R Statistical programming language, R Core Team (2017). The data could also be directly downloaded from our github site.

We demonstrate some examples using the software. Figure (10) shows the predictions for three materials: $Ba_{0.2}La_{1.8}CuO_4$, MgB_2, and Hg. The "verbose" option uses the cosine similarity measure to pull data with similar chemical formulas. The multiple entries for $Ba_{0.2}La_{1.8}CuO_4$ are obtained. The default value for verbose is false so no superconductors similar to MgB_2 and Hg are shown.

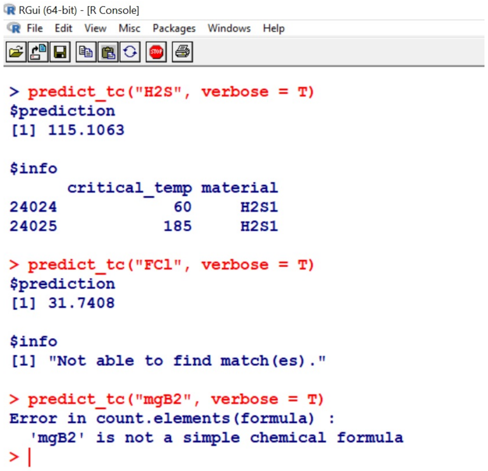
We had obtained the data on July 24, 2017. We like to see what sort of predictions we could obtain for some new superconductors reported since. Nishiyama *et al.* (2017) report a T_c of around 3 K for $Ca_{0.5}Sr_{0.5}C_6$. Goto *et al.* (2017) report a T_c of 1.3 K for $NaSn_2As_2$. Figure (11) shows the prediction results. The XGBoost model over-predicts but it is within the ±9.5 K out-of-sample rmse. The message "Not able to find match(es)" indicates that nothing in the training data is similar to these two new superconductors. We should not expect good predictions for completely new superconductors.

Figure (12) shows what can go wrong when the XGBoost model predicts badly or when the inputs do not make sense. The prediction for H_2S, which has a T_c of 203 K under extremely high pressures, is way off. This is perhaps expected since there is no feature that captures the dependence of T_c on pressure. The model gives a prediction for FCl but this is a non-sense; The prediction model can't check for the existence of solids. The model gives an error message for mgB2


RGui (64-bit) - [R Console]
File Edit View Misc Packages Windows Help

```
> 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)."

> |
```

Figure 11: This figure shows the software prediction results for $Ca_{0.5}Sr_{0.5}C_6$ and $NaSn_2As_2$ which have reported critical temperatures of 3 K and 1.3 K respectively.


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
> |
```

Figure 12: This figure shows the software prediction results for H_2S, and (non-sense) FCl, and misspelled formula mgB2.

since it does not recognize mg with the lower case m as an element.

5 Bibliography

References

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. `https://arxiv.org/abs/1603.02754`.

Chen, T., He, T., Benesty, M., Khotilovich, V., and Tang, Y. (2018a). *xgboost: Extreme Gradient Boosting*. R package version 0.6.4.1.

Chen, X., Huang, L., Xie, D., and Zhao, Q. (2018b). Egbmmda: Extreme gradient boosting machine for mirna-disease association prediction. *Cell Death and Disease*, **9**(3).

Conder, K. (2016). A second life of the matthiass rules. *Superconductor Science and Technology*, **29**(8).

Dick, J. M. (2008). Calculation of the relative metastabilities of proteins using the chnosz software package. *Geochemical Transactions*, **9**(10).

Freund, Y. (1995). Boosting a weak learning algorithm by majority. *Information and Computation*, **121**(2), 256 – 285.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Ann. Statist.*, **29**(5), 1189–1232.

Goto, Y., Yamada, A., Matsuda, T. D., Aoki, Y., and Mizuguchi, Y. (2017). Snas-based layered superconductor nasn2as2. *Journal of the Physical Society of Japan*, **86**(12), 123701.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning, Data Mining, Interference, and Prediction*. Springer, 2 edition.

Izenman, A. J. (2008). *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*. Springer, 1 edition.

Nishiyama, S., Fujita, H., Hoshi, M., Miao, X., Terao, T., Yang, X., Miyazaki, T., Goto, H., Kagayama, T., Shimizu, K., Yamaoka, H., Ishii, H., Liao, Y.-F., and Kubozono, Y. (2017). Preparation and characterization of a new graphite superconductor: Ca0.5sr0.5c6. *Scientific Reports*, **7**(7436).

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Sheridan, R. P., Wang, W. M., Liaw, A., Ma, J., and Gifford, E. M. (2016). Extreme gradient boosting as a method for quantitative structureactivity relationships. *Journal of Chemical Information and Modeling*, **56**(12), 2353–2360. PMID: 27958738.

Valentin, S., Oses, C., Kusne, A. G., Rodriguez, E., Paglione, J., Curtarolo, S., and Takeuchi, I. (2017). Machine learning modeling of superconducting critical temperature. `https://arxiv.org/abs/1709.02727`.

Wolfram and Research (2017). Mathematica, version 11.2.

A Mathematica ElementData

Below is the list of sources Mathematica has used to obtained the element property data. It is directly copied from:
http://reference.wolfram.com/language/note/ElementDataSourceInformation.html.

- Atomic Mass Data Center. “NUBASE.” 2003. http://amdc.in2p3.fr/web/nubase_en.html
- Cardarelli, F. Materials Handbook: A Concise Desktop Reference. Springer, 2000.
- Lide, D. R. (Ed.). CRC Handbook of Chemistry and Physics. 87th ed. CRC Press, 2006.
- Speight, J. Lange’s Handbook of Chemistry. McGraw-Hill, 2004.
- United Kingdom National Physical Laboratory. “Kaye and Laby Tables of Physical and Chemical Constants.” http://www.kayelaby.npl.co.uk/
- United States National Institute of Standards and Technology. “Atomic Weights and Isotopic Compositions Elements.” https://www.nist.gov/pml/atomic-weights-and-isotopic-compositions-relative-atomic-masses
- United States National Institute of Standards and Technology. “NIST Chemistry Webbook.” http://webbook.nist.gov/chemistry/
- Winter, M. ”WebElements.” 2007. https://www.webelements.com/