Regression of Superconducting Critical Temperature: using a PCA-GridSearch-Adaboost Regression Model

Naresh Aketi
Department of Computer Science and Engineering,
JNTUACEP, India
pandu5188@gmail.com

Harsha Praneeth Dussa
Department of Computer Science and Engineering,
JNTUACEP, India
harshapraneeth10@gmail.com

Suresh Parachuri
Department of Computer Science and Engineering,
JNTUACEP, India
psuresh0597@gmail.com

Hanmanthu Uppara
Department of Computer Science and Engineering,
JNTUACEP, India
Hanumanth557chandu@gmail.com

Abstract -

Superconductivity is being studied since its discovery more than a century ago. The numerous applications of the superconductors made it a subject of intense research. Despite being studied for so long, some of its properties remain a mystery. One of the interesting properties of a super conductor is its critical temperature. Superconductors exhibit zero electrical resistance when maintained at the critical temperature. The value of critical temperature is different for each superconducting material. This value is experimentally calculated by measuring resistance against the temperature of the material. In this project, by taking advantage of the immense increase of readily accessible and potentially relevant information, we develop several machine learning methods modeling critical temperature of a super conductor based on its chemical properties. The final model will give an estimate of critical temperature of a superconductor. This estimate provides confidence on a newly discovered material to continue further research on it.

Keywords – Superconductivity, Critical temperature, Adaboost regression, Machine learning, Decision tree regression, Supervised learning model.

I. Introduction

As important functional materials, high-transition temperature (high-TC) superconductors have some typical physical parameters, such as transition temperature Tc, magnetic susceptibility and critical current density (Jc), which make them very useful in many practical applications like magnetically levitated trains and power transmission. Previous researches showed that the high-Tc superconductors are generally characterized by a two-dimensional layered superconducting condensate with unique features that are not traditional superconducting metals. Their important property, Tc, is determined by their layered crystals, bond lengths, valency properties of the ions, and Coulomb coupling between electronic bands in adjacent, spatially separated layers.

It is clear that Tc (critical temperature) of superconductors depend on its other chemical properties. In this project we utilize the already available data about superconductors to estimate the critical temperature of new potential materials. The developed model can be used to gain confidence on a new material to conduct further research on it regarding its superconducting behavior. Also, experimental determination of critical temperature is a laborious process which is made easy when an estimate of the value is provided by our model.

II. Proposed Work

The goal is to create a regression model to predict the critical temperature of a super conductor. Tasks involved in the project implementation are as follows:

- Gathering, analyzing and preprocessing the data (data exploration)
- Set a benchmark model and evaluation metric
- Training different regression models
- Tuning the final model

Figure 2.1: Machine learning work flow

The final model is expected to give an estimate of critical temperature of a super conductor based on its chemical properties. And this is the architecture of a machine learning model,

A. Gathering and preprocessing data

The data set is taken from the UCI data repository at:

https://archive.ics.uci.edu/ml/datasets/Superconductivty+Data

The characteristics of the dataset are:

Data Set Characteristics:	Multivariate	Number of Instances:	21263
Attribute Characteristics:	Real	Number of Attributes:	81
Associated Tasks:	Regression	Missing Values?	N/A

Table 2.1: Characteristics of the dataset

Since working with too many features puts so much burden on the learning model, a feature extraction method is employed to see that the number features can be reduced.

	number_of_elements	mean_atomic_mass	wtd_mean_atomic_mass	gmean_atomic_mass	wtd_gmean_atomic_mass	entropy_atc
number_of_elements	1.000000	-0.141923	-0.353064	-0.292969	-0.454525	
mean_atomic_mass	-0.141923	1.000000	0.815977	0.940298	0.745841	
wtd_mean_atomic_mass	-0.353064	0.815977	1.000000	0.848242	0.964085	
gmean_atomic_mass	-0.292969	0.940298	0.848242	1.000000	0.856975	
wtd_gmean_atomic_mass	-0.454525	0.745841	0.964085	0.856975	1.000000	
entropy_atomic_mass	0.939304	-0.104000	-0.308046	-0.190214	-0.370561	
wtd_entropy_atomic_mass	0.881845	-0.097609	-0.412666	-0.232183	-0.484664	
range_atomic_mass	0.682777	0.125659	-0.144029	-0.175861	-0.352093	
wtd_range_atomic_mass	-0.320293	0.446225	0.716623	0.458473	0.673326	
std_atomic_mass	0.513998	0.196460	-0.060739	-0.121708	-0.274487	
wtd_std_atomic_mass	0.546391	0.130675	-0.089471	-0.166042	-0.331657	
mean_fie	0.167451	-0.285782	-0.209296	-0.367690	-0.276668	
wtd_mean_fie	0.484445	-0.222097	-0.522595	-0.354664	-0.612317	
gmean_fie	0.024229	-0.240565	-0.109490	-0.286844	-0.154323	

Table 2.2: Correlation values of the attributes

After doing the PCA the 80 features in the input data are projected to 3 features and the total variance explained by these 3 dimensions is 97.22%. Any new dimensions would not contribute much in terms of explained variance and this is evident by the graph below,

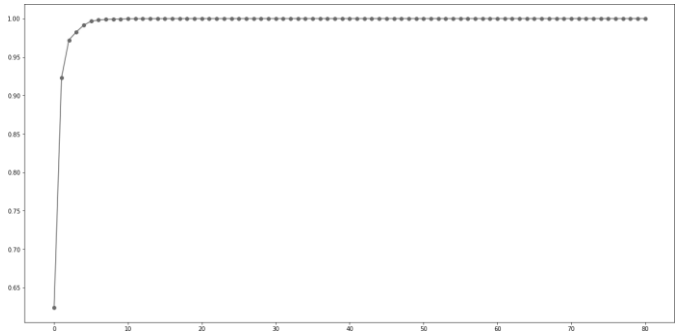

Figure 2.2: Explained variance values against the dimensions of the dataset

The new transformed data after doing principal component analysis looks like,

	Dimension 1	Dimension 2	Dimension 3
0	-5184.9320	-3.5949	-11.3954
1	-5272.0599	-45.6752	-175.9628
2	-5293.8237	-56.3574	-217.1071
3	-4662.6669	172.0064	-591.5800
4	-5174.6743	-58.8022	380.8708

Table 2.3: New transformed data

B. Benchmark model and Evaluation metric

R2_score (Coefficient of determination) is a common metric for a regression model; it is a statistical measure of how well the regression predictions approximate the real data points.

$$R^2 \equiv 1 - \frac{SS_{res}}{SS_{tot}}$$

The benchmark r2_score, training time and predicting time are created based on the linear regression model's performance. Linear regression is the simplest regression model. As the Occam's Razor suggests going with the simplest model, it is first checked whether the data follows a linear trend or not.

C. Algorithms

The decision process is taken by choosing the criteria or attribute with highest entropy value and tree is constructed until the threshold of minimum entropy reaches. Finally, the leaf nodes are the classes in this case the estimate of the output variable.

This is how a decision tree works,

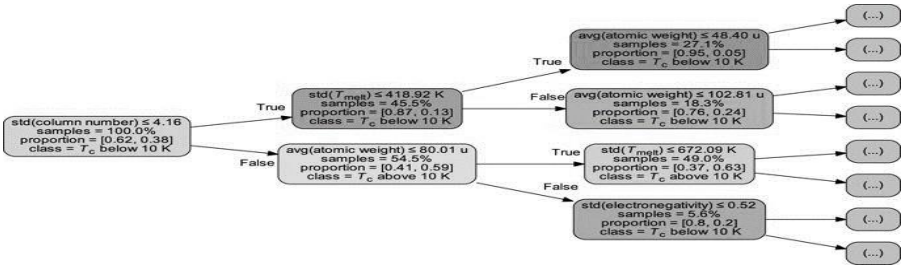

Figure 2.4: Decision tree working

Random forest generates trees randomly and those trees are essentially weak leaners. The algorithm combines the weak learners to produce the end result. This approach of random tree generation can be replaced with a comparatively new and efficient ensemble method called adaboost.

The random forest algorithm's working is explained in the below diagram,

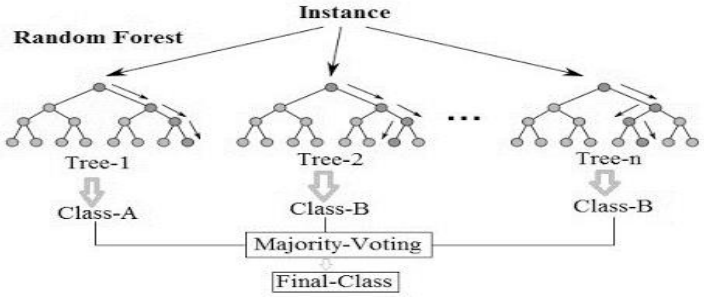

Figure 2.5: Working of Random forest model

Adaboost generates trees which will perform well on the areas its predecessors couldn't. The adaboost model performs better than other primitive models so its performance is compared to other models and chosen to be the best model for this particular regression task.
The working description of an adaboost algorithm is shown below,

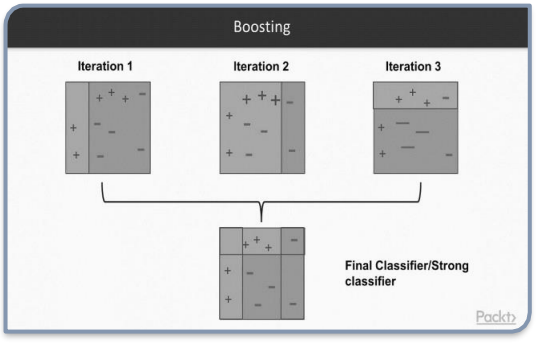

Figure 2.6: Adaboost working

D. *Training models*

An initial look at the output variable suggests that it mimics stepwise data. Amongst the different regression models the decision tree regression model comes first to mind when dealing with step wise data. So initially a decision tree model is trained to fit the data.

The results after performing the decision tree regression are as follows,

DecisionTreeRegressor

{

‘train_time’: 0.17719674110412598 seconds,

‘pred_time’: 0.00800633430480957 seconds,

‘score_train’: 0.7845250110933649,

‘score_test’: 0.7835929355566275

}

After the training the decision tree model the results were moderate in terms of fitting the data with 78% r2_score on the data. But the model does very well when it comes to training and prediction time. This suggests that a more powerful model can be used for this regression task.

Adaboost generates trees which will perform well on the areas its predecessors couldn’t. A Decision tree regressor is passed to the adaboost algorithm to boost it. Then all the estimators are combined to produce the best results and here they are,

AdaboostRegressor

{

‘train_time’: 1.5947909355163574 seconds,

‘pred_time’: 0.20222735404968262 seconds,

‘score_train’: 0.9684746229251533,

‘score_test’: 0.9671931187916423

}

E. Tuning the final model

The final model must be tuned to improve its performance and one way to do so is using a grid search method. In grid search the parameters passed to the model are tuned by using cross validation. The parameters of the adaboost model which are tuned are

- Number of estimators
- Max depth of the tree

The result of the grid search is as follows,

AdaBoostRegressor(

base_estimator = DecisionTreeRegressor(criterion=’mse’, max_depth=25, max_features=None,

max_leaf_nodes = None, min_impurity_decrease = 0.0,

min_impurity_split = None, min_samples_leaf = 1,

min_samples_split = 2, min_weight_fraction_leaf = 0.0,

presort = False, random_state = None, splitter = ‘best’),

learning_rate = 1.0, loss = ‘linear’, n_estimators = 30,

random_state = None) trained.

Through grid search the best parameters for the adaboost model are found out to be, Number of estimators is 30 and maximum depth is 25. With this fine tuned model, we can produce the best result possible.

Result Analysis

The final model is evaluated using the R2_score. The input data is split into train and test dataset. The train dataset is used to train the model and then the test dataset is used to see how the model does on previously unseen data.

A. Models summary

The summary of the performance of all models is,

Model name	R2_score
Linear regression	30%
Decision tree regression	78%
Random forest regression	92%
Adaboost regression	96%

Table 3.1: models summary

Since, the data doesn't have a linear trend, it is expected that the linear regression model doesn't perform well. So, it ended with 30% r2_score. The data is in a stepwise trend, so the decision tree model does pretty well with a 78% r2_score. The improvement for a simple decision tree model is an ensemble method. Random forest is introduced to improve the r2_score of the decision tree model. With the random forest model, the r2_score is increased to 92%. The final model choice would be Adaboost which is also a ensemble method. The best it could do is 96%.

B. Final model analysis

It is clear that the adaboost outclasses other models in terms of r2_score. But this adaboost is further improved through grid search the final improved model's performance is noted as follows,

train_time: 0.4915473461151123

pred_time: 0.1772010326385498

score_train: 0.982052608960896

score_test: 0.9833312158619923

The final model is seen to do well with 98.33% r2_score, that means the final model can explain 98% of variance in the data. To see the final model in work, some actual and predicted values are,

	critical_temp	pred_ct
0	7.700	6.600000
1	6.930	4.500000
2	1.489	1.104667
3	2.150	3.950000
4	17.980	17.990000
5	65.000	65.000000
6	8.100	4.700000
7	12.000	11.150000
8	7.750	4.650000
9	6.000	6.000000

Table 3.2: Comparison of actual and predicted values

The actual and predicted values are quite near, this means the model works well. To take a deeper look a plot is drawn on actual and predicted values.

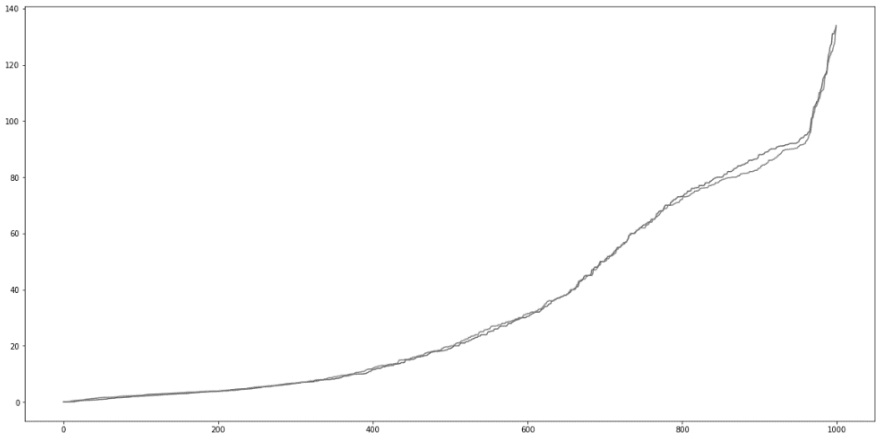

Figure 3.1: Plot of actual and predicted values

The two curves are almost overlapped suggesting that the model is performing well by predicting values near to actual values.

III. Conclusion

A PCA-PSO-ADA model for predicting TC from structural and correlative electronic parameters of high-TC superconductors. Adaboost was adopted to deal with the dataset, which was a small sample set, and the PSO (Grid search) algorithm was utilized to search for its optimal parameters to achieve a good performance. The PCA was employed to reduce dimensions and interdependencies between the parameters, and the selected optimal dimensions of the parameters were subsequently utilized in PSO-Adaboost to train and validate the regression model. According to the assessment results and comparison, the PCA-PSO-ADA model provided a better accuracy of prediction than the other models for the dataset. At last, additional data was used to validate the prediction, and the results were also reasonable. In a word, machine-learning methods can be applied to some domains of materials and the PCA-PSO-ADA ensemble method may be used to predict the TC of new high-TC superconductors.

The final the adaboost model estimates the critical temperature of the super conductor with 98% explained variance, which mean it would give a very good estimate.

IV. Future enhancement

- To further improve the predictive power of the models, another set of features can be constructed based on crystallographic and electronic information
- By building a model which can extract the required features for our model from different combinations of elements it might be possible to find new superconductors
- Also instead of using the PCA to reduce the dimensionality, when a powerful configuration of firmware is available, all the features can be used to train the estimator.

References

[1] Hirsch, J. E., Maple, M. B. & Marsiglio, F. Superconducting materials: conventional, unconventional and undetermined. Phys. C. 514, 1–444 (2015).

[2] Anderson, P. W. Plasmons, gauge invariance, and mass. Phys. Rev.130, 439–442 (1963).

[3] Chu, C. W., Deng, L. Z. & Lv, B. Hole-doped cuprate high temperature superconductors. Phys. C. 514, 290–313.

[4] Bergerhoff, G., Hundt, R., Sievers, R. & Brown, I. D. The inorganic crystal structure data base. J. Chem. Inf. Comput. Sci. 23, 66–69 (1983).

[5] Agrawal, A. & Choudhary, A. Perspective: materials informatics and big data: realization of the "fourth paradigm" of science in materials science. APL Mater. 4, 053208 (2016).

[6] Seko, A., Maekawa, T., Tsuda, K. & Tanaka, I. Machine learning with systematic density-functional theory calculations: application to melting temperatures of single- and binary-component solids. Phys. Rev. B 89, 054303–054313 (2014).

[7] Curtarolo, S. et al. AFLOWLIB.ORG: a distributed materials properties repository from high-throughput ab initio calculations. Comput. Mater. Sci. 58, 227–235 (2012).

[8] Villars, P. & Phillips, J. C. Quantum structural diagrams and high-Tc superconductivity. Phys. Rev. B 37, 2345– 2348 (1988).

[9] Ling J., Hutchinson M., Antono E., Paradiso S., and Meredig B. High-dimensional materials and process optimization using data-driven experimental design with well-calibrated uncertainty estimates. Integr. Mater. Manuf. Innov. 6, 207–217 (2017).

[10] Hirsch, J. E. Correlations between normal-state properties and superconductivity. Phys. Rev. B 55, 9007–9024.

[11] Ward, L., Agrawal, A., Choudhary, A. & Wolverton, C. A general-purpose machine learning framework for predicting properties of inorganic materials. NPJ Comput. Mater. 2, 16028 (2016).

[12] Yang, K., Oses, C. & Curtarolo, S. Modeling off-stoichiometry materials with a high-throughput ab-initio approach. Chem. Mater.28, 6484–6492 (2016).