Composition of concrete and its influence on compressive strength

Filipe P. de Farias
Department of Teleinformatics Engineering
Federal University of Ceará
Fortaleza, Brazil
filipepfarias@fisica.ufc.br

Yvo J. M. Sales
Department of Teleinformatics Engineering
Federal University of Ceará
Fortaleza, Brazil
yvo@gtel.ufc.br

***Abstract*—The compressive strength of a concrete is a important property since it impacts directly on its applications. The classical approach to obtain the compressive strength of a specific concrete mixture is to submit a sample to a test on a hydraulic press. However, it takes time to perform this type of test since it is necessary to wait the sample to cure. In this work, we try to find a regression model to accurately estimate the compressive strength of a concrete mixture from the concentration of its components.**

***Index Terms*—Regression models, concrete compressive strength, machine-learning, partial least squares.**

I. Introduction

A material formed by aggregates bonded together by a fluid material that hardens over time has been used by humans for construction since many years ago [3]. Nowadays this material is known as concrete and it's widely used in the construction field. The aggregates used in the mixture of the concrete affect directly its compressive strength which highly impacts its applications. For instance, in general, the concrete for columns or beams needs to have a greater compressive strength than the one for pavement. On the previous work, we have made a statistical analysis on a dataset extracted from the UCI Machine Learning Repository (University of California, Irvine) [4] that collects information about the concentration of some aggregates used to form different mixtures of concretes. In this work, we try to find a regression model to estimate the relation between the concentration of those aggregates and the strength of the concrete mixture. The goal is that such model could be a good replacement to tests of samples on hydraulic press.

This work is divided as follows. A description of the data is given in Section II resulted from the previous work with the addition of the regressor (concrete compressive strength). Section III brings a brief introduction to the regression models that will be used to fit the data. In the sequel, we present and discuss the results in Section V. Finally, the conclusions and considerations are exposed in Section VI.

II. Data description

The composition of each one of the N = 1030 concrete samples is given by the concentrations (kg/m^3) of D components: Cement, Blast Furnace Slag, Fly Ash, Water, Superplasticizer, Coarse Aggregate and Fine Aggregate, as summarized in Table I. The cement is what binds the elements of the concrete together. Indeed his technical name in the literature is *binder* [6]. The other components as blast furnace slag and fly ash, the outcomes of another industrial process reused in the concrete mixture, they have the role of increase the chemical hardness of the concrete, i.e. in a microscopic level. The water is responsible for react with the cement resulting in the cement stone. The superplasticizer gives fluid characteristics to the concrete aiming to better fill the mold and decrease the use of water. The coarse and fine aggregates give some macroscopical mechanical resistance to the concrete but can reduce its compressive strength if bad applied. Their major role is to occupy the spaces in the mold reducing the use of cement. The output is the concrete compressive strength which is measure in the stress test where a force its applied to a sample using a hydraulic press. When the sample reaches the rupture point, the pressure, force per area of the sample, is observed.

TABLE I
Data description

Label	Component	Unit
D_1	Cement	kg/m^3
D_2	Blast Furnace Slag	kg/m^3
D_3	Fly Ash	kg/m^3
D_4	Water	kg/m^3
D_5	Superplasticizer	kg/m^3
D_6	Coarse Aggregate	kg/m^3
D_7	Fine Aggregate	kg/m^3
D_8	Age	days
$\mathbf{D_9}$	**Compressive strength**	**MPa**
Total	N = 1030 samples	

The concrete mixtures were divided into a set $\mathcal{L} = \{L_1, L_2, L_3\}$ of classes [1] based on their compressive strength, following the function $\mathcal{C}\colon \mathcal{R} \mapsto \mathcal{L}$ defined in Eq. 1. The mixture which is weak and not recommended for structures, the *Non-standard*, was labeled with L_1 and comprises 295 samples. The mixture whose strength is in a range that can be applied to structures is classified as L_2, or *Standard*, and comprises 525 samples. The high performance mixture

L_3, *High-strength* and comprises 210 samples.

$$\mathcal{C}(D_9) = \begin{cases} L_1\,, D_9 < 25 \\ L_2\,, 25 \leq D_9 < 50 \\ L_3\,, D_9 \geq 50 \end{cases} \tag{1}$$

where D_9 is the compressive strength of the concrete mixture.

The observations are the measured compressive strengths of each sample and, as the predictors $D_1 - D_7$, are real valued. The Age (D_8) of the concrete is extremely discrete. At Fig. 1 we show the separation between the classes by their means and variances. These predictors were chosen because they are the ones with the major correlation with the compressive strength, in absolute terms, as presented in the previous work. We note that this separation is not clear, then another techniques being needed, using all the predictors in order to try to separate the classes.

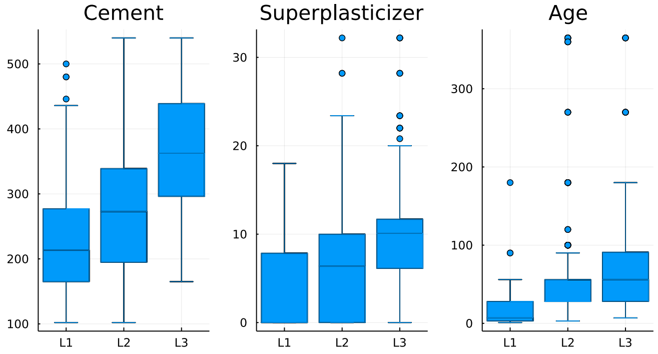

Fig. 1. Boxplot of each class for the concentration of cement, superplasticizer and the time of cure (age).

III. Regression models

Regression models try to find relations between the *independent variables* and the *dependent variables*, which are named, respectively, predictors and outcomes in this work. These relations can occur in different forms. The simplest one is the linear relationship, which is when the curve predictors *vs* outcomes, in the case that both are one-dimensional, forms a simple line and, in the general case, a hyperplane. Continuing the discussion of the last work, we now change the strategy to use a non-linear method for regression, in this case using neural networks.

A. Neural networks regression

In this method, expressed in Eq.2, each predictor x_k is weighted by a real-valued constant w_k. The outcome is inputed in an activation function ϕ which will be "activated" when that sum be enough to reach a given output Y. This output will be the data, specifically the value of the compressive strength.

$$Y = \phi\left(\sum_k w_k \cdot x_k\right) \tag{2}$$

The shape of ϕ is a choose of the one who is modelling. Some of them are the sigmoid and the step functions. In this work, the ReLU function was used, which is defined as

$$\phi_{\text{ReLU}}(x) = \max(0, x). \tag{3}$$

The next step, which is to optimise a cost function is quite similar to the ordinary linear squares, but now observing that the function is different, but yet the cost will be defined as the distance between the data and the values of the model.

IV. Classification models

The regression models are used to fit output variables that are quantitative. But, in the majority of the applications, the output variables are qualitative instead, which means that classification models must be used to fit those variables. The classification models estimate the probability of the outcome Y to belong to a class L given a set of predictors X, i.e. the probability $Pr(Y = L|X)$.

In this section, we briefly describe the classification models that were used throughout this paper.

A. Linear discriminant analysis

The linear discriminant analysis (LDA) models the probability distribution of the predictors for each one of the outcome classes and applies the Bayes' theorem to them in order to estimate the probabilities $Pr(Y = L|X = x)$, which is the probability of the outcome Y to belong to the class L given that the predictors X assume a value x. Let π_L be the prior probability of a random chosen sample to belong to the class L and $f_L(x) = Pr(X = x, Y = L)$ denote the probability density function of x given that a sample belongs to the class L, then the Bayes' theorem states that

$$Pr(Y = L|X = x) = \frac{\pi_L f_L(x)}{\sum_{i=1}^{|\mathcal{L}|} \pi_i f_i(x)}\,. \tag{4}$$

For two classes, the LDA classifies the samples by finding a line that best segregate them. For more than two classes, the LDA works on a similar way but now it looks for multiple lines that each one segregates the samples between two classes per time.

B. Support vector machine classifier

The support vector machine (SVM) classifier constructs a set of hyperplanes that can be used to segregate the samples into classes. A hyperplane is chosen when it has the largest distance to the nearest training-data point of any class. When the data is not linearly separable, a kernel function is used to transform it and then the SVM classifier is applied to the transformed data. Fig. 2 illustrates a SVM application for a two dimensional case.

C. K-nearest neighbors

The k-nearest neighbors (KNN) classifier designates the class of a sample as the one associated to the majority of its k nearest neighbors. Fig 3 illustrates an example of the use of the KNN classifier with $k = 3$.

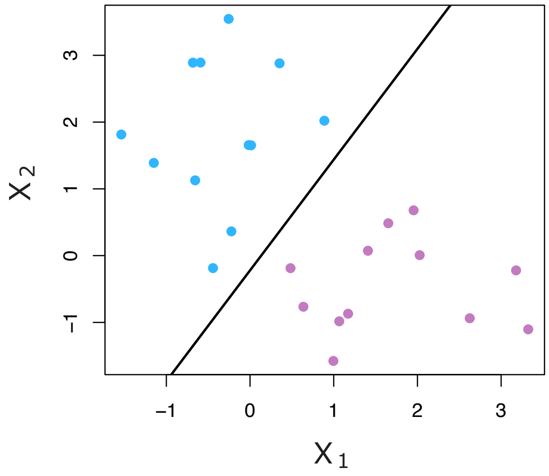

Fig. 2. Hyperplane dividing the samples into two classes. Source: extracted from [7].

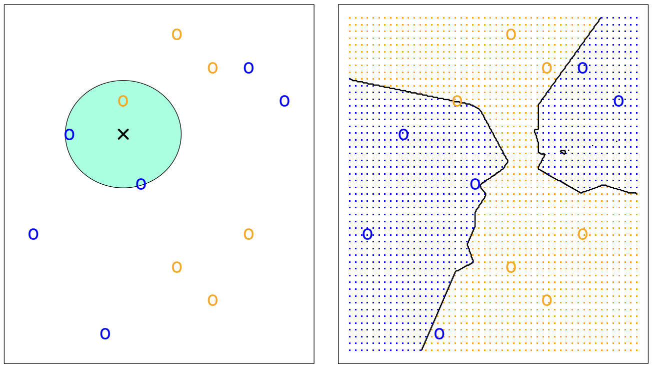

Fig. 3. Example of the KNN classifier with $k = 3$. Left: illustration of a classless sample and its neighborhood. Right: illustration of the boundary for the data considering this KNN classifier. Source: extracted from [7].

TABLE II
Confusion matrix for the KNN classifier

		True Values		
		L_1	L_2	L_3
Predicted	L_1	292 (28.3495%)	0 (0.0%)	0 (0.0%)
	L_2	2 (0.19%)	523 (50.8%)	2 (0.16%)
	L_3	1 (0.097%)	2 (0.19%)	208 (20.2%)

TABLE III
Performance metrics for the KNN classifier

Class	Accuracy	Precision	Recall	F1 score
L1	0.997087	1.0	0.989831	0.994889
L2	0.994175	0.99241	0.99619	0.994297
L3	0.995146	0.985782	0.990476	0.988124

TABLE IV
Confusion matrix for the SVM Classifier

		True Values		
		L_1	L_2	L_3
Predicted	L_1	149 (14.5%)	23 (2.2%)	0 (0.0%)
	L_2	146 (14.2%)	491 (47.7%)	169 (16.4%)
	L_3	0 (0.0%)	11 (1.1%)	41 (4.0%)

TABLE V
Support Vector Machine for Classification

Class	Accuracy	Precision	Recall	F1 score
L1	0.835922	0.866279	0.505085	0.638116
L2	0.661165	0.609181	0.935238	0.737791
L3	0.825243	0.788462	0.195238	0.312977

V. Results

The results were obtained from the models with neural networks regression and k-nearest neighbors after 5-fold cross validation. For KNN, was obtained that for $k = 5$ we have the performances at the Tab. II and Tab. III. We can observe that the KNN method, we obtain 99% of the accuracy, which represents a low rato of false classifications among the true ones. The 98% of recall represents that the the method barely loses a true positive classification. To finish, it was obtained about of 99% of F_1-score which is the harmonic mean between the precision and the recall.

TABLE VI
Confusion matrix for the LDA classifier

		True Values		
		L_1	L_2	L_3
Predicted	L_1	240 (23.3%)	143 (13.9%)	1 (0.1%)
	L_2	49 (4.8%)	277 (26.9%)	56 (5.4%)
	L_3	6 (0.6%)	105 (10.2%)	153 (14.9%)

TABLE VII
Performance metrics for the LDA classifier

Class	Accuracy	Precision	Recall	F1 score
L1	0.806796	0.625	0.813559	0.706922
L2	0.657282	0.725131	0.527619	0.610805
L3	0.836893	0.579545	0.728571	0.64557

From the “more non-linear” method, the SVM for classification, to the linear one, the LDA, for classification, it was observed a decrease in the performance. This is shown on the Tab. IV and Tab. V for the SVC and in Tab. VI and Tab. V for LDA. In accuracy both the methods are similar, differing on the misclassification rates as the recall. An important remark is that the SVM for classification has space for improvement since the 5-fold cross validation was not performed in order to optimise the hyperparameters of the kernel function (in this case the squared exponential parameters) to better fit the data. This feature is the great advantage of this method giving him a greater flexibility. The LDA method by the way, is already the optimal in a Bayes’ sense.

TABLE VIII
NEURAL NETWORK REGRESSION

CV	**RMSE**(MPa)	R^2
1-th fold	612.62	0.294586
2-th fold	612.04	**0.355655**
3-th fold	**611.87**	0.261329
4-th fold	612.92	0.243543
5-th fold	612.59	0.276319

In Tab. VIII it's presented the results for the neural networks regression. Only about 35% of the variance of the data was comprised by the model, which is unsatisfactory beyond its mean squared error of 611.87 MPa, on the best scenario. Given the poor concrete is about 25 MPa, the error is huge for the purpose of using the model in the case.

VI. Conclusions

It was expected by these authors that the neural network regression could outperform the linear regression, what didn't occur. This is one of the challenges encountered in the modelling of this dataset. Despite of that, when the data is stratified in an attempt to classify, at least, if the concrete sample is good for basic applications as the classes L_2 and L_3 are, the models were capable of classify well. Highlighting the performance of the k-nearest neighbors, which achieve 99% of F_1-score. Then it is possible to say that this non-linear method outperform the linear one, LDA.

References

[1] ACI Manual of Concrete Practice 2000, Part 1: Materials and General Properties of Concrete. American Concrete Institute. Farmington Hills, MI.

[2] Tibshirani, Robert, et al. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Germany, Springer New York, 2009.

[3] Mindess, S., and Young, J.F. Concrete. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1981.

[4] I-Cheng Yeh, "Modeling of strength of high performance concrete using artificial neural networks," Cement and Concrete Research, Vol. 28, No. 12, pp. 1797-1808 (1998)

[5] I-Cheng Yeh, "Prediction of Strength of Fly Ash and Slag Concrete By The Use of Artificial Neural Networks," Journal of the Chinese Institute of Civil and Hydraulic Engineering, Vol. 15, No. 4, pp. 659-663 (2003).

[6] Khasanov, Irmuhamedova, et al, "Theoretical foundations of the structure formation of cement stone and concrete," IOP Conf. Series: Materials Science and Engineering 869 (2020)

[7] James, G., Witten, D., Hastie, T. and Tibshirani, R., 2013. An introduction to statistical learning (Vol. 112, p. 18). New York: springer.