

Chapter 6

NovaCast: Prediction of alloy parameters

6.1 Introduction

Daniel Gillblad

NovaCast produces software for the foundry industry, including tools for solidification simulation and thermal analysis. These tools are used for metallurgical process control to achieve less casting defects, a high yield and reduced costs.

Thermal analysis studies the solidification from liquid metal into solid iron or an alloy. It is based on recording temperatures at certain time intervals during the solidification progress, and from them constructing a cooling curve. The cooling curve is essentially a plot of the temperature of the metal as a function of time.

By thermal analysis of a sample from the furnace, it is possible to extract information that can be used to predict properties of the alloy produced by the contents of the furnace. In the thermal analysis, a sample from the furnace is cooled and the cooling curve is recorded. Several parameters can then be extracted from the curve, describing important properties of the cooling process. The parameters include, among other things, plateau temperatures and cooling rates in the different states. Using this, information about when different state transitions occur in the furnace can be extracted, which in turn gives an opportunity to predict properties such as chemical composition and final quality of the alloy.

Predicting properties of the final alloy from a sample taken from the furnace is an important task. The ability to make such predictions reliably could potentially help in the reduction of scrap material and defects in the foundry. There are two completely separate NovaCast data sets. For the first data set, the task was to estimate the number of graphite nodules per mm^2 . For the second data set, which contains more variables and several alloy properties, the main focus of the work within the DALLAS project has been on predicting one of these properties, the oxygen content.

6.1.1 The first NovaCast data set

The first NovaCast dataset consists of measurements from 96 different furnace samples. For each sample, the following attributes are available:

1. TL, Liquidus temperature in the cooling curve.
2. TES, Start eutectic cooling.
3. TEU, Lower eutectic temperature.

Name	Explanation	Remarks
TL	Liquidus temperature in the cooling curve	
TES	Start eutectic cooling	
dT/dTES	Cooling rate at the start of eutectic solidification	
TE Low	Lower eutectic temperature	
TE High	Upper eutectic temperature	
R	Recalescence	
Max Rate	Max R rate	
T1	Temperature 1	
T2	Temperature 2	
T3	Temperature 3	
TL Plata	Liquidus temperature plateau	Only in ladle data
dT/dtTS	Cooling rate at the solidus temperature	
TS	Solidus temperature	
GRF2	Grafite factor 2	

Table 6.1: The attributes included in all measurements except the tellurium cup.

4. TEH, Upper eutectic temperature.
5. GRF1, Grafite factor 1.
6. dT/dt.TS, Cooling rate at the solidus temperature.
7. GRF2, Grafite factor 1.

There are two attributes that are interesting to predict:

8. MICROSHR, Micro suction tendency.
9. NOD_COUNT, The nodule count, the number of graphite nodules per mm^2 .

All input variables are continuous. Of the output variables, the nodule count is a continuous variable and the micro suction tendency categorical.

6.1.2 The second NovaCast data set

The second dataset consists of measurements from two different places in the process, denoted furnace data and ladle data. The furnace data set contain 45 samples, and the ladle data 46 samples. The datasets have been treated as completely separate. Although both data sets contain roughly the same attributes, they must be regarded as behaving significantly different from each other. There are six different kinds of measurements in both data sets:

1. Grey unioculated
2. 12mm cup
3. Grey inoculated
4. Tellurium cup
5. Second grey unioculated
6. Second 12mm cup

Name	Explanation
TL	Liquidus temperature in the cooling curve
TES	Eutectic temperature

Table 6.2: The attributes included in the tellurium cup.

The first and second 12mm cups and the first and second grey unicolated measurements are duplicate samples, and should be highly correlated. All of the different kinds of measurements except the tellurium cup include the attributes listed in table 6.1 along with short explanations of some of them. The tellurium cup measurements contains just 2 attributes. These attributes are listed in table 6.1. All the attributes are continuous.

The data also contains thirty attributes that might be interesting to predict. All the these possible output attributes are continuous, and describe for example chemical composition and hardness. The most important output attribute to predict is the oxygen content.

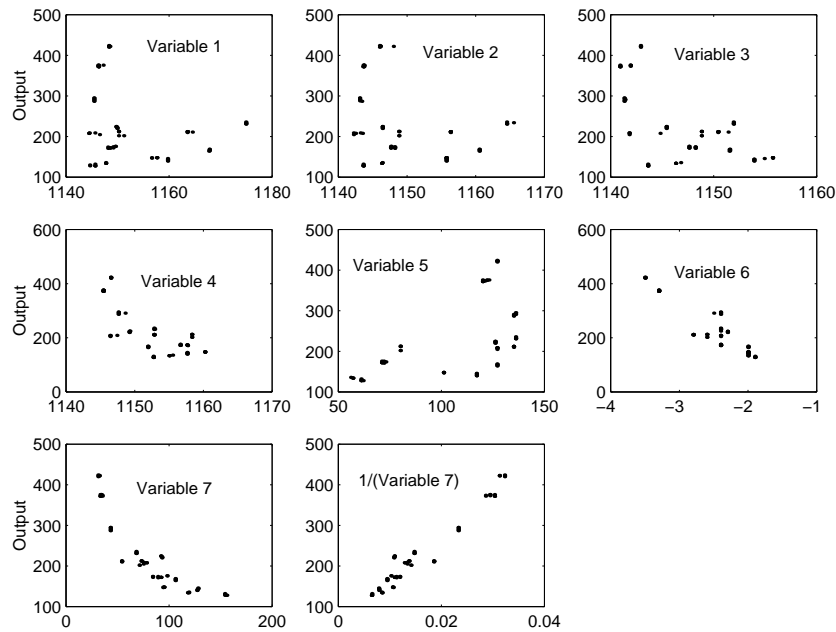


Figure 6.1: Scatter plots of the output versus the seven input variables. It is obvious that $1/x_7$ is better to use in the modeling than x_7 itself.

6.2 The model used at Halmstad University Thorsteinn Rögnvaldsson

6.2.1 Data, preprocessing, and variable selection

The first NovaCast data set had two parts, of which we only considered one, namely the regression problem of estimating the number of graphite nodules per mm^2 . The data set is very limited (few variables and few observations).

For the first data set, NovaCast were interested in what variables that are important for the prediction, how good predictions can be made, and if there were dependencies among the variables.

The data set consists of 96 observations, with 7 possible input variables for each observation, and two outputs, of which we only focus on one. Of these, 11 were set aside for a blind out-of-sample test after the model construction, and the remaining 85 were used for training.

6.2.2 Preprocessing

We first studied how the input variables varied with the output, which is shown in Figure 6.1. From this we decided to invert variable 7, which improved the correlation with the output. A cross correlation study, see Figure 6.2, showed that variables 1 and 2 were strongly linearly correlated, as well as variable 6 with the inverse of variable 7. This could mean that one of the variables should be removed or they should be replaced by composite variables. However, the exhaustive search variable selection method we used (see below) meant that we did not have to care too much about this.

We also looked for outliers in the data, but did not see any observations that were out of the ordinary.

The variables were preprocessed in the following way

1. The seventh variable was inverted (*i. e.* $x_7 \rightarrow 1/x_7$).
2. All variables were normalized to zero mean and unit standard deviation.

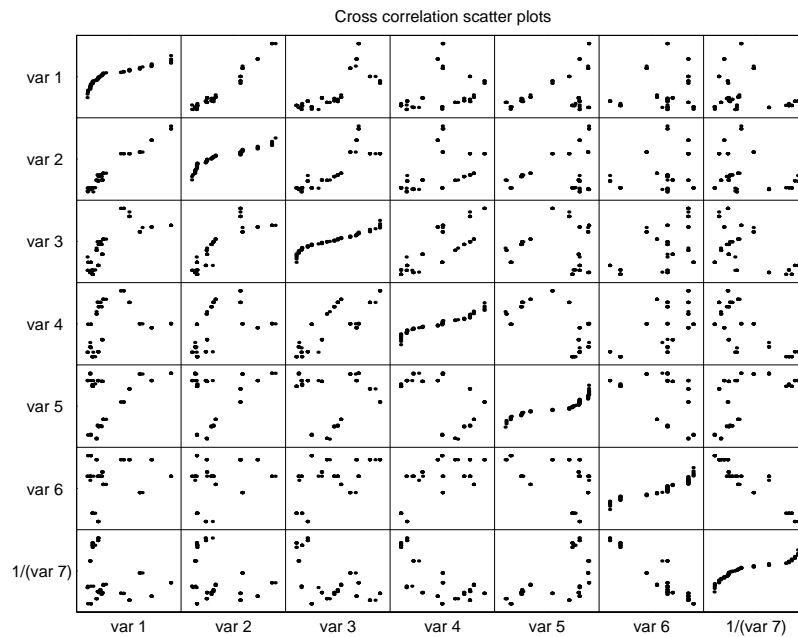


Figure 6.2: Scatter plots of the input variables versus each other. Variables 1 and 2 are positively linearly correlated. Variable 6 and the inverse of variable 7 are negatively linearly correlated. The plots along the diagonal are normal probability plots for the variable in question.

6.2.3 Variable selection

The number of variables was very small and the number of observations was also very small. We therefore tried all possible combinations of input variables to the simple models (*i. e.* those that were quick to construct). The total number of possibilities is given by

$$N_{poss} = \sum_{k=1}^K \binom{K}{k} \quad (6.1)$$

which equals 127 for $K = 7$. Each variable set is evaluated by computing the 5-fold cross validation error for the model.

In the multilayer perceptron case, where the training time is considerable, we instead selected variables by backwards elimination. That is, we started with using all 7 variables and then removed the variable that lead to the largest decrease (or smallest increase) of the cross validation error.

6.2.4 Constructing the regression models

We tried three different regressors: Linear regression, k -nearest neighbors (k NN) regression, and the multilayer perceptron (MLP).

The linear regressor was constructed by using the pseudoinverse method. Each linear model, defined by the input variable set, was evaluated using the 5-fold cross validation error (*i. e.* $127 \times 5 = 635$ linear models were constructed and tested).

The k NN regressor is constructed by trying $k = \{1, 2, 3, 4, 5, 6, 7, 8\}$ and all possible combinations of input variables. Each combination of k and input variable set was evaluated using the 5-fold cross validation error (*i. e.* a total of $127 \times 8 \times 5 = 3810$ models were constructed and tested). Euclidean metric was used throughout for all k NN models.

The MLP regressor was trained using the Levenberg-Marquardt optimization and early stopping. We used 3/5 of the data for computing the gradient, 1/5 of the data for determining the stopping point, and 1/5 of the data for validating after training. The number of hidden units was varied between 3 and 12. The five best models trained for the best variable set were combined into an averaging committee (the committee members have the same number of hidden units, and the same input variables). In summary, we trained $7 \times 10 \times 5 = 350$ MLP models and tested them.

6.2.5 Results

The best linear model turned out to be a model with all seven input variables. The cross validation mean square error (CV-MSE) for this model was 285 ± 95 (the latter number is the cross validation estimated standard deviation). The best k NN model used only variables 7 and 2, with $k = 1$ (*i. e.* only one nearest neighbor). The CV-MSE for this model was 4.2 ± 0.9 . A k NN model with $k = 1$ is equivalent to a look up table with no interpolation. The best MLP model used seven hidden units and variables 5, 6, and 7. The CV-MSE for this model was 4.7 ± 2.5 . An MLP using all input variables and seven hidden units gives $\text{CV-MSE} = 8.3 \pm 13$, for comparison.

These models were then tested on the hold-out test data set. The result is summarized in Tables 6.3 and 6.4. The outputs from each model is plotted versus the true output in Figure 6.3.

Model	MSE	RMS	RMS%	ρ
Linear (all variables)	140.5	11.8	0.4%	0.9900
k NN ($k = 1, 2$ variables)	2.7	1.7	0.008%	0.9998
MLP (7 hidden, 3 variables)	2.0	1.4	0.007%	0.9999

Table 6.3: Summary of results on the hold-out test set (11 samples). The RMS% column shows how many percent wrong the model is on average. The ρ column shows the correlation coefficient (Pearson's correlation coefficient) for the prediction versus the true value. The precision of the numbers sets a theoretical lower limit for MSE at 0.25.

Model	Linear	k NN	MLP
Linear		-	-
k NN	+		0
MLP	+	0	

Table 6.4: The significance of the results, tested using the Mann-Whitney test. The test tests whether the residuals from two models come from the same distribution. The linear model is significantly poorer than both the k NN and MLP models, at the 95% significance level. The k NN and MLP results are not significantly different, at a 95% significance level.

6.2.6 Conclusion

The results illustrates that a k NN model is often a strong competitor to methods like MLP. The k NN model is both simple to construct, easily modified, nonparametric, and nonlinear. It is often a good idea to use a k NN model as benchmark for another nonlinear method, k NN being a well established statistical method with a very appealing simplicity.

In this case the MLP gave slightly better results (not significant though) on the hold-out test set. However, the results on the cross validation data during the training phase indicated that the k NN method should have been the better one of the two.

The input data had strong correlations, which indicated that partial least squares (PLS) would have been a better linear benchmark than simple least squares.

A bigger data set is needed to really compare models and test whether the important variables have really been extracted.

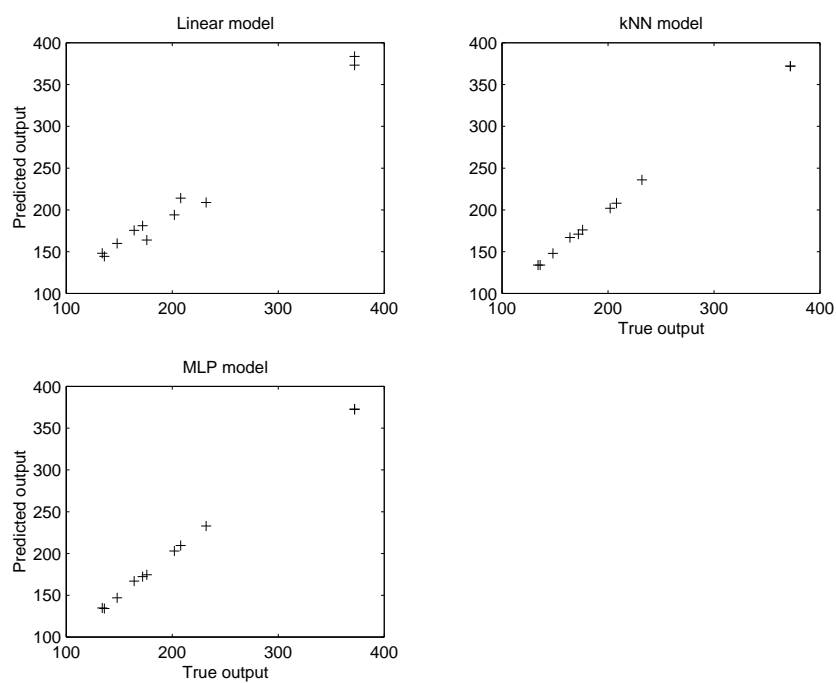


Figure 6.3: Scatter plots for the predicted output versus the true output for the hold-out test data set.

6.3 The model used at SICS

Daniel Gillblad

6.3.1 Oxygen content prediction using a mixture model

A natural approach to predict a continuous parameter, in this case the oxygen content, is to create a mixture of Gaussians over the whole input space and the output space. The mixture model parameters, such as the means and the covariance matrices of the Gaussians are estimated from the training data set, using for example expectation maximization (EM). The marginal of the output space given an input pattern can then be calculated. When we know the marginal distribution of the output variables, the expectation or the maximum of this distribution can be used as a prediction, depending on whether we want to minimize the mean square error or the absolute error.

The performance of a mixture model over all input attributes and the output attribute were tested. All the tests used the furnace data set, and all possible input attributes were used, a total of 68. The number of available training samples were 44. The expectation of the resulting marginal distribution for a test pattern was used as the prediction. Table 6.5 a and b show the test results, table a showing the results when the model was tested on training data and table b with leave one out cross-validation. With leave one out cross-validation, a model is estimated from all entries in the data except one, which the model is tested on. This is done for all patterns in the data set. The first column in the tables shows the number of Gaussians used in the mixture. The second column the resulting root mean square error (RMS), and the third and fourth column show the fraction of patterns that are within one and three standard deviations of the predicted pattern. The third and fourth column can be viewed as a measure of how many of the predictions that are, in one sense, reasonable. The standard deviation of the oxygen content is 0.26.

The results are reasonably good, both on training and test data. The mean square error is rather low, at least for some of the tested models. It is obvious from the test results that while using only one Gaussian, the mean square error is rather high. On the other hand, for the results with cross validation, the number of patterns within one standard deviation is high, suggesting that a simple linear predictor might be sufficient to produce good results if this is considered to be the most important property. Increasing the number of Gaussians though leads to lower mean square errors, and the number of predictions that fall within three standard deviations rise up to 100lower. This probably makes for a more practically useful prediction.

Note that the results both on the training set and with cross-validation are not completely consistent, in the respect that the quality of the results do not follow the number of Gaussians in a very organized way. This is due to random effects. When generating the prediction model, the initial model that is trained is generated at random with some considerations to the data.

6.3.2 Dependency structure analysis

To gain insight and knowledge of a data set, a dependency structure analysis can be very useful. The dependency graph will show the dependencies between attributes, and what attributes that affects the prediction the most. Even if not used for predictions or some other application, the creation of a dependency graph can still be very valuable for getting a feel for the relationships in the data.

When constructing the dependency graph, we need to keep in mind what we want to use it for. By calculating all pairwise correlations between attributes and then showing the strongest ones in a dependency graph, we might get useful information about the general dependency structure and what attributes that might be redundant. On the other hand, dependencies that might be interesting for a certain task might not be visible using this approach. Often the strongest dependencies in data are between similar input attributes, not between input attributes and the output attribute we are interested in. If we want to visualize the dependencies to a specific output attribute, we must use another approach.

The dependency graph in figure 6.4 was generated keeping the relevant output attribute, the oxygen content, in mind. All pairwise linear correlations, *i. e.* the correlation coefficients, between attributes

4 a. Training and testing on all data				
Number of Gaussians	Root mean square error (RMS)	Within one standard deviation	Within three standard deviations	
1	0.35	70.5%	88.6%	
2	0.20	77.2%	100.0%	
3	0.23	86.4%	100.0%	
4	0.20	77.2%	100.0%	
5	0.17	86.4%	100.0%	
6	0.14	88.6%	100.0%	
7	0.18	77.2%	100.0%	
8	0.51	81.8%	100.0%	

4 b. Leave one out cross-validation				
Number of Gaussians	Root mean square error (RMS)	Within one standard deviation	Within three standard deviations	
1	0.57	77.2%	84.1%	
2	0.35	52.2%	95.5%	
3	0.32	59.1%	95.5%	
4	0.40	86.4%	95.5%	
5	0.32	65.9%	95.5%	
6	0.30	50.0%	97.7%	
7	0.26	61.4%	100.0%	
8	0.27	61.4%	100.0%	

Table 6.5: Oxygen content prediction results on furnace data

in the ladle data were calculated. Then the strongest correlations between input attributes and the oxygen content were selected, as well as all the stronger correlations between these input attributes. All other correlations were discarded. The graph in the figure was then generated by running a greedy tree construction algorithm on the selected dependencies. The attributes in the graph are denoted with the attribute name and a number in parenthesis describing from what kind of measurement the attribute belongs to (see section 6.1.2). When studying figure 6.4, keep in mind that the oxygen content has a strong correlation to all the input attributes in the graph. The tree structure shown is most of all an aid to understand the relationship between the input variables.

Two general observations can be made by examining figure 6.4. First, most input attributes shown belong to either Grey unioculated or Second grey unioculated. This means that these measurements might contain more useful information about the oxygen content than the other measurements. To be fair though, the grey inoculated is also rather common while the tellurium cup and 12mm cup is barely represented. Second, most attributes are temperatures or the maximum or minimum value of temperatures. This is perhaps not surprising since most of the inputs in fact represent different temperatures, but for example TEHigh and TELow are clearly over represented, indicating that these attributes might be important for the oxygen content.

Also note that after the tree generation, the oxygen content ended up as a leaf, with only one connection. This is a result of the fact that the dependencies between the input attributes are generally stronger than to the oxygen content.

On the whole, though, the dependency graph must be looked upon with some scepticism. The number of examples are low, only 44, while at the same time the data must be considered to be fairly noisy. This means that reliable correlation estimation might be hard, and is also the reason why the simple correlation coefficient was used instead of some more advanced measure more prone to suffer from the low number of examples available. Also the tree generation algorithm used is very sensitive to random effects and noise in the correlation estimations, but still it can provide some useful information as a suggestion of what the dependency structure might look like.

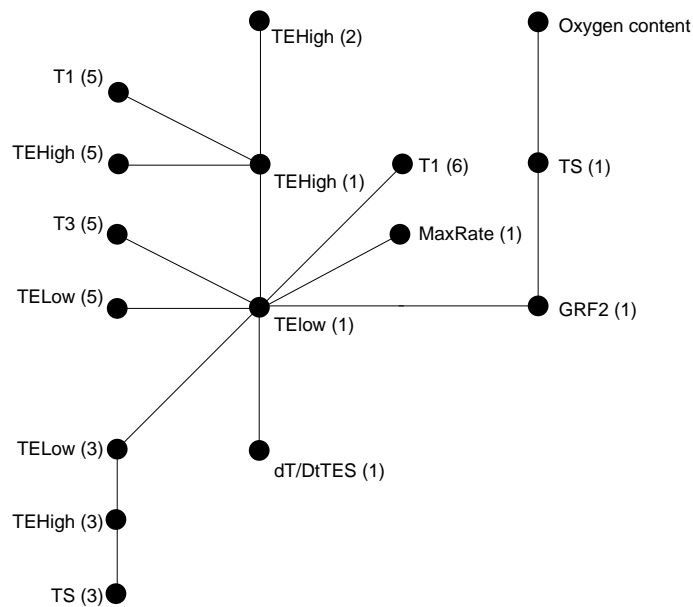


Figure 6.4: Dependency graph for the ladle data

6.3.3 Comments and conclusions

The predictions produced by the very straightforward method of using a Gaussian mixture model over the whole input and output space are promising. The results are reasonably good, and there is probably room for significant improvement using similar but more specialized models. The data set is very small, though, and it is hard to tell whether this dataset correctly reflects all the properties of the data.