World Automation Congress © 2010 TSI Press.

OVERCOMING DATA GATHERING ERRORS FOR THE PREDICTION OF MECHANICAL PROPERTIES ON HIGH PRECISION FOUNDRIES

JAVIER NIEVES, IGOR SANTOS AND PABLO G. BRINGAS
S[3]Lab
University of Deusto
Bilbao, Spain
{javier.nieves, isantos, pablo.garcia.bringas}@deusto.es

ABSTRACT—Mechanical properties are the attributes of a metal to withstand several loads and tensions. More accurately, *ultimate tensile strength* (UTS) is the force a material can resist until it breaks. The only way to examine this feature is the use of destructive inspections that render the casting invalid with the subsequent cost increment. In our previous researches we showed that the foundry process can be modelled as an expert knowledge cloud to anticipate the value of the UTS with outstanding results. Nevertheless, the data gathering phase for the training of machine learning classifiers is performed in a manual manner. In this paper, we present the use of *Singular Value Decomposition* (SVD) and *Latent Semantic Analysis* (LSA) with the aim of reducing the number of ambiguities and noise in the dataset. Furthermore, we have tested this approach comparing the results without this pre-processing step in order to illustrate the effectiveness of the proposed method.

Key Words: fault prediction, machine-learning, industrial processes optimization.

1. INTRODUCTION

Foundry is one of the axes of current economy: thousands of castings are manufactured in foundries around the world to be part of more complex systems, say for instance, brake of a car, propeller of a boat, wing of an aircraft or the trigger in a weapon. As one may think, the tiniest error may have fatal consequences and, therefore, if one of the pieces is found faulty, this fact may be detrimental to both individuals and for businesses activities. Moreover, current trends encourage the production of smaller and more accurate components. It is really easy to produce castings and suddenly discover that every single one is faulty. Commonly, the techniques for the assurance of failure-free foundry processes are exhaustive production control and different simulation techniques [1].

Besides, a huge amount of production-process-related factors may influence, harm or diminishes the mechanical properties of an iron casting. Likewise, there are several defects that may appear within it. In this paper, we focus on the so-called *ultimate tensile strength* (UTS) that is the force a casting can withstand until it breaks or, in other words, it is the maximum stress any material can withstand when subjected to tension. Therefore, manufactured iron castings have to reach a certain value (e.g. threshold) of UTS in order to pass the strict quality tests. Unfortunately, the only way to examine the UTS breaks the piece and thus it incurs a cost increment.

As shown in [2, 3], a machine-learning-based tool could help in this goal. Still, there were some irregularities on the data that yield the result not as effective as it should. More accurately, the data we work with has several records that appear incorrect. The reason why these inconsistencies appear is because the data acquisition is performed in a manual fashion. One solution to this issue is to provide a more accurate data-gathering system; still, the hard conditions of the foundry process itself yield this task as very difficult. Instead, we provide a method that is able to reduce noise in data.

Thereby, Latent Semantic Analysis (LSA) has been used in document retrieval [4, 5] with successful results in reducing noise and ambiguities in the training dataset. Regarding pre-processing tasks for industrial prediction, one of the most recent works is the one from Pham et al. [6], that uses a *bee algorithm* (i.e. similar to genetic algorithms) in order to perform a more comprehensive feature selection for support vector machine training. Nevertheless, the accuracy increment their approach achieves, a 10.93 of increment, is lower than ours, a 16.99 of increment.

Against this background, this paper advances the state of the art in two main ways. First, we describe LSA as a data pre-processing step to the machine-learning-based UTS prediction system. Second, we evaluate this approach with real raw data obtained from a real foundry process in order to compare the accuracy and suitability of this method with the previous ones [2, 3].

2. MECHANICAL PROPERTIES AND FOUNDRY PRODUCTION

Several factors contribute to render the foundry process very complex, such as the extreme conditions in which it is developed. Thereby, this process has to go through numerous phases, however, when it refers to iron ductile castings, these phases can be simplified in the followings. First, the *melting and pouring phase* where the raw metals are melt, mixed and poured onto the sand shapes. Second, the *moulding phase* where the moulding machine forms and prepares the sand moulds. And finally, the *cooling phase* where the solidification of the castings is controlled in the cooling lines until this process is finished.

In this way, after all phases, the obtained castings are subject to forces (loads). In that moment, engineers calculate these forces and how the material behaves under several conditions. Specifically, the most important mechanical properties of foundry materials are the following ones [7]: strength (the UTS is a specific kind of strength), hardness**,** toughness, resilience, elasticity, plasticity**,** brittleness, ductility and malleability.

Furthermore, there are several procedures for testing the value of mechanical properties of the materials in a laboratory, like ASTM standards [8]. Unfortunately, the only way for discovering the level of these properties is the employment of destructive inspections. Moreover, the process requires suitable devices, specialised staff and quite a long time to analyse the materials. For example, in UTS testing process, a little part of the original casting, called specimen, is placed on a testing machine that pulls it measuring how is changing the tensile strength before breaking it. More accurately, the mechanical properties are related with several variables [9, 10] and, consequently, they allow us to predict these properties. Hence, we should take them into account in order to design our machine-learning models (in our case, we use 25 variables to carry out our experiments).

3. LATENT SEMANTIC ANALYSIS

As aforementioned, our foundry dataset suffers from noise in dataset instances. Latent Semantic Analysis overcomes this problem by using statistically-derived concepts instead of singular features for machine-learning. It uses truncated Singular Value Decomposition (SVD) [11] to transform a high dimensional vector into a lower-dimensional semantic vector, by projecting the former into a semantic subspace.

Thereby, suppose the rank of the dataset matrix A is r, SVD decomposes A into the product of three matrices. Firstly, matrix U describes the original row entities as vectors of derived orthogonal factor values. Secondly, matrix V describes the original column entities in the same way. Finally, a diagonal matrix $\sum$ containing scaling values. When the three components are multiplied, the original matrix is reconstructed $(A = U\Sigma V^T)$, where $U = (u_1, u_2, ..., u_n) \in R^{t\times r}$, $\Sigma = diag(\sigma_1, \sigma_2, ..., \sigma_n) \in R^{t\times r}$. V^T is the transpose of V. σ_i 's are A 's singular values, $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r$. U and V are column-orthonormal. Furthermore, the columns of U and V are the left and right singular vectors, respectively, corresponding to the monotically decreasing (e.g. in value) diagonal elements of $\sum$ which are called the singular values of the matrix A. LSI approximates A with a rank-k matrix

$$A_k = U_k \Sigma_k V_k^T \tag{1}$$

by omitting all but the k largest singular values, where $U_k = (u_1, u_2, ..., u_n)$, $\Sigma_k = diag(\sigma_1, \sigma_2, ..., \sigma_n)$, $V_k = (v_1, v_2, ..., v_k)$. Row i of $U_k \in R^{t\times k}$ is the representation of feature i in the k-dimensional semantic space. An instance vector $q \in R^{t\times 1}$ can be folded into the k-dimensional semantic space applying equation $q_k = U_k^T q$ or $q_k = \Sigma_k^{-1} U_k^T q$. Their difference is whether to scale the vector by the inverse of the singular values. Regarding LSA, A_k is the closest k-dimensional approximation to the original term-document space represented by the incidence matrix A. As stated previously, by reducing the dimensionality of A, much of the “noise” that causes poor retrieval performance is thought to be eliminated. Therefore, despite a high dimensional representation seems to be required for a good retrieval performance, care must be taken not to reconstruct A. If A is nearly reconstructed, the noise caused by variability of feature choice and features that span or nearly span the data collection will not be eliminated, resulting in poor performance.

4. MACHINE-LEARNING CLASSIFIERS

4.1 Artificial neural networks (ANN)

ANN is a machine learning model that simulates the behaviour of neurons in the human brain [12]. Formally, a neuronal network consists on interconnected neurons. The activation of a neuron depends on its set of inputs, where y_i is the activation of the current neuron, f_i is the activation function, $W_{j,i}$ is the weight of the neuron and a_j is the activation of the input neuron: $y_i = f_i(\sum_{j=1}^{n} W_{j,i} \cdot a_j)$.

More accurately, *multilayer perceptron* (MLP) is a kind of artificial neural network model of simple neurons called perceptrons that are structured in layers classified as *input layers*, *hidden layers* and *output layer*. We perform the training of the model using *backpropagation algorithm* [12] that calculates the weights $W_{j,i}$ of the activation function for each neuron.

4.2 Bayesian Networks

"Bayes' theorem" is the basis of the so-called Bayesian inference. In this way, this theorem adjusts the probabilities as new information on evidences appears. According to its classical formulation:

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)} \tag{2}$$

given two events A and B, the conditional probability $P(A \mid B)$ that A occurs if B occurs can be obtained if we know the probability that A occurs, $P(A)$, the probability that B occurs, $P(B)$, and the conditional probability of B given A, $P(B \mid A)$.

Bayesian networks are probabilistic models for multivariate analysis. Formally, they are directed acyclic graphs associated to a probability distribution function [13]. Nodes in the graph represent variables, and the arcs represent conditional dependencies between such variables. Further, the probability function illustrates the strength of these relationships in the graph [13].

4.3 Decision Trees

These classifiers [14] constitute a decision support tool represented as a tree-like graph. Their internal nodes (decision nodes and chance nodes) are tests regarding the problem's variables and their final nodes or leaves are the final decision of the algorithm.

Moreover, there are several training algorithms that are typically used for learning the graph structure of these trees using a labelled dataset. In this work, we used random forest, which is an ensemble different randomly-built decision trees. Besides, we also used J48 (the Weka [15] implementation of the C4.5 algorithm).

4.4 K-Nearest Neighbours (KNN)

This classifier is one of the simplest supervised machine-learning algorithms for classifying instances [16]. In its training phase, represents a set data instances $S = s_1, s_2, ..., s_n$ in a *n*-dimensional space where *n* is the amount of variables for each instance.

On the other hand, the classification phase is developed by measuring the distance between the training instances and the unknown instance. In this way, the distance between two points can be calculated using any distance measure, in our case we used Euclidean distance $\sqrt{\sum_{i=1}^{n}(X_i - Y_i)^2}$. Finally, one of the most used techniques is to classify the unknown instance as the most common class amongst the *K-nearest neighbours*.

4.5 Support Vector Machine (SVM)

SVM [17] consists on finding a hyperplane that divides the *n*-dimensional space of the data in two regions. This hyperplane is the one that maximises the *margin* between those two regions. Specifically, this maximal margin is defined by the largest distance between the examples of the two classes.

Supporting vectors are the instances that are situated near the hyperplane. Since sometimes the space cannot be divided with a hyperplane, a kernel function K is used. This function studies the relations within the data and creates complex divisions in the space.

5. EXPERIMENTS

We have collected data from a foundry specialised in safety and precision components for the automotive industry, principally in disk-brake support with a production over 45000 tons a year.

These experiments are focused exclusively in the UTS prediction. Note that, this experiment can be extrapolated to the prediction of other mechanical properties. Furthermore, the only way to examine the mechanical properties is in *a posteriori* fashion and carrying out destructive inspections. Moreover, according to the very restrictive quality standards imposed by clients, pieces flawed with an invalid value of UTS must be rejected.

In these experiments, the machine-learning models have been built with the aforementioned 25 variables. We have worked with 11 different references (i.e. type of pieces) and we have used as input data the results of the destructive inspection from 889 castings (note that each reference may involve several castings or pieces) performed in beforehand. To this extent, we have defined two risk levels: Risk 0 (more than 370 *MPa*) and Risk 1 (less than 370 *MPa*).

Specifically, we have followed the next configuration for the performed experiment:

- **Latent semantic analysis:** We have built a dataset with the result of applying Latent Semantic Analysis to the original dataset with the aim of comparing the results of the machine-learning classifiers with and without this pre-processing step.
- **Cross validation:** In order to obtain a proper representation of the data, *K-fold cross validation* is usually used in machine-learning evaluation [12]. In our experiments, we have performed a K-fold cross validation with *K=10.*
- **Learning the model:** For each fold, we have performed the learning phase of each algorithm with the corresponding training dataset, applying different parameters or learning algorithms depending on the model. More accurately, we have applied the following models:
    - *Bayesian networks:* For Bayesian networks we have used different structural learning algorithms: K2, Hill Climber, Tree Augmented Naïve (TAN) and Naïve Bayes Classifier.
    - *Artificial neural networks:* We have used a three-layer Multilayer Perceptron (MLP) learned with *backpropagation* algorithm.
    - *Support Vector Machines:* For *SVM* we have performed experiments with a polynomial kernel, a normalised polynomial Kernel, Pearson VII function-based universal kernel and a radial basis function (RBF) based kernel.
    - *K-nearest neighbour:* For *K-nearest neighbour* we have performed experiments with $k = 1$, $k = 2$, $k = 3$, $k = 4$ and $k = 5$.
    - *Decision Trees:* We have performed experiments with J48, the *Weka* implementation of the *C4.5* algorithm, and Random Forest, an ensemble of randomly constructed decision trees.
- **Testing the model:** For each fold, we have evaluated the percent of correctly classified instances and the area under the ROC curve that establishes the relation between false negatives and false positives.

6. RESULTS

Table I shows the results in terms of percent of correctly classified instances. In this way, we can notice that from the 16 tested classifiers, 15 of them obtained a statistical significant improvement when applying Latent Semantic Analysis. In this way, Support Vector Machines with Pearson VII Function Based Universal Kernel outperformed the rest of the classifiers with an accuracy of 97.74%, which is a spectacular result increasing more than 16 points the achieved accuracy level. On the other hand, *Radial Basis Function kernel* for Support Vector Machines had a significant decrease of accuracy.

Dataset	***Without Pre-processing***	***Applying LSI***	
SVM with Pearson VII Function based Universal Kernel	80.75	97.74	✓
SVM with Polynomial Kernel	82.07	97.72	✓
Multilayer Perceptron	82.19	97.24	✓
SVM with Normalised Polynomial Kernel	83.78	96.91	✓
Decision Tree: Random Forest $(n = 50)$	86.65	94.42	✓
Naïve Bayes	75.07	91.55	✓
Bayesian Network with TAN	79.57	89,74	✓
Decision Tree: J48	81.66	89.21	✓
Bayes Network with K2	77.20	88.86	✓
Bayes Network with Hill Climber	77.78	88.86	✓
KNN $(K = 1)$	82.64	88.63	✓
KNN $(K = 5)$	81.52	87.69	✓
KNN $(K = 3)$	81.15	87.06	✓
KNN $(K = 4)$	80.96	85.60	✓
KNN $(K = 2)$	78.84	84.36	✓
SVM with RBF Kernel	81.71	72.55	×

✓, ×, - statistically significant improvement, degradation or not either statistically significant improvement or degradation

Table I. Results in Terms of Accuracy

Likewise, Table II shows the results in terms of area under the ROC curve. To this extent, from the 16 classifiers that we have tested, 15 achieved a significant improvement after applying Latent Semantic Analysis. In the same way as in the accuracy results, Radial Basis Function kernel for support vector machines had a significant decrease of accuracy. This time, Artificial Neural Network and Random Forest outperformed the rest of the classifiers with more or less an area of 1. Anyway, the rest of classifiers achieve good results, they are between 0.97 and 0.87.

Summarizing, the results validate our hypothesis that by applying *Latent Semantic Analysis* the system is capable of reducing the noise in the dataset. In this way, the excellent results we have obtained yield this approach to be deployed in the production process of a real foundry.

Dataset	***Without Pre-processing***	***Applying LSI***	
Multilayer Perceptron	0.85	1.00	✓
Decision Tree: Random Forest $(n = 50)$	0.92	0.99	✓
Naïve Bayes	0.82	0.97	✓
SVM with Polynomial Kernel	0.74	0.96	✓
SVM with Pearson VII Function based Universal Kernel	0.66	0.96	✓
Bayesian Network with TAN	0.86	0.95	✓
Bayes Network with K2	0.84	0.95	✓
Bayes Network with Hill Climber	0.84	0.95	✓
SVM with Normalised Polynomial Kernel	0.75	0.95	✓
KNN $(K = 5)$	0.85	0.93	✓
KNN $(K = 4)$	0.85	0.93	✓
KNN $(K = 3)$	0.86	0.92	✓
KNN $(K = 2)$	0.84	0.91	✓
Decision Tree: J48	0.76	0.88	✓
KNN $(K = 1)$	0.79	0.86	✓
SVM with RBF Kernel	0.72	0.50	×

✓, ×, - statistically significant improvement, degradation or not either statistically significant improvement or degradation

Table II. Results in Terms of ROC Area

7. CONCLUSIONS

Foreseeing the mechanical properties in ductile iron castings is one of the hardest challenges in foundry-related research. In addition, with these great results, we can provide some profits to high precision foundries like savings by the reduction in their fault rates.

In this work, we focus on the pre-processing method used for noise reduction in the data in order to improve the current prediction of ultimate tensile strength. In this way, we have compared the results of this approach with the previous one, showing that this new approach provides better results and is able to handle noise in the training dataset.

The future development of this predictive tool is oriented in three main directions. First, we plan to extend our analysis to the prediction of other defects in order to develop a global network of incident analysis. Second, we will compare more supervised and semi-supervised machine learning algorithms in order to prove their effectiveness to predict foundry defects. Finally, we plan to integrate the best classifiers in a meta-classifier combining the partial results.

REFERENCES

1. J. Sertucha, A. Loizaga, and R. Suárez, "Improvement opportunities for simulation tools," in *Proceedings of the 16th European Conference and Exhibition on Digital Simulation for Virtual Engineering*, 2006, invited talk.
2. J. Nieves, I. Santos, Y. K. Penya, S. Rojas, M. Salazar, and P. G. Bringas, "Mechanical properties prediction in high-precision foundry production." in *Proceedings of the 7th IEEE International Conference on Industrial Informatics (INDIN 09)*, 2009, pp. 31–36.
3. I. Santos, J. Nieves, Y. K. Penya, and P. G. Bringas, "Machine-learning-based mechanical properties prediction in foundry production," in *In Proceedings of ICROS-SICE International Joint Conference (ICCAS-SICE)*, 2009, pp. 4536–4541.
4. I. Matveeva, G. Levow, A. Farahat, and C. Royer, "Term representation with generalized latent semantic analysis," *Amsterdam Studies in the Theory and History of Linguistic Science Series 4*, vol. 292, pp. 45, 2007.
5. S. Sakellaridi, H. Fang, and Y. Saad, "Graph-based multilevel dimensionality reduction with applications to eigenfaces and latent semantic indexing," in *Proceedings of the 2008 Seventh International Conference on Machine Learning and Applications-Volume 00*. IEEE Computer Society, 2008, pp. 194–200.
6. D. Pham, Z. Muhamad, M. Mahmuddin, A. Ghanbarzadeh, E. Koc¸, and S. Otri, "Using the bees algorithm to optimise a support vector machine for wood defect classification," in *Memorias del Innovative Production Machines and Systems Virtual Conference*, 2007.
7. C. W. Lung and N. H. March, "Mechanical Properties of Metals: Atomistic and Fractal Continuum Approaches". *World Scientific Pub Co Inc*, July 1992.
8. A. S. for Testing and Materials, "ASTM B489 - e1 Standard Practice for Bend Test for Ductility of Electrodeposited and Autocatalytically Deposited Metal Coatings on Metals," 2008.
9. R. Gonzaga-Cinco and J. Fernández-Carrasquilla, "Mechanical properties dependency on chemical composition of spheroidal graphite cast iron," *Revista de Metalurgia, vol. 42,* pp. 91–102, March–April 2006.
10. J. Fernández-Carrasquilla and R. Ríos, "A fracture mechanics study of nodular iron," *Revista de Metalurgia*, vol. 35, no. 5, pp. 279–291, 1999.
11. G. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numerische Mathematik*, vol. 14, no. 5, pp. 403–420, 1970.
12. C. M. Bishop, "Neural Networks for Pattern Recognition". *Oxford University Press*, 1995.
13. E. Castillo, J. M. Gutiérrez, and A. S. Hadi, "Expert Systems and Probabilistic Network Models". *Springer-Verlag*, 1997.
14. J. Quinlan, "Induction of decision trees," *Machine learning, vol. 1, no. 1,* pp. 81–106, 1986.
15. S. Garner, "Weka: The Waikato environment for knowledge analysis," in *Proceedings of the New Zealand Computer Science Research Students Conference*, 1995, pp. 57–64.
16. E. Fix and J. L. Hodges, "Discriminatory analysis: Nonparametric discrimination: Small sample performance," *Technical Report Project 21-49-004, Report Number 11*, 1952.
17. V. Vapnik, "The nature of statistical learning theory". *Springer*, 2000.