A comparison of wood log dissimilarities to predict sawmill output with k-Nearest Neighbor algorithms

Sylvain Chabanet, Mathis Dumas, Hind Bril El-Haouzi, Philippe Thomas

4rd International Conference on Advances in Signal Processing and Artificial Intelligence (ASPAI' 2022), 19-21 October 2022, Corfu, Greece

Type of Presentation:
Oral: ☒ In-person: ☒
Poster: ☐ Virtual in Zoom: ☐
The same: ☐

Topic:
Machine Learning, applied artificial intelligence

A comparison of wood log dissimilarities to predict sawmill output with k-Nearest Neighbor algorithms

S. Chabanet [1], M. Dumas [1] H. Bril El-Haouzi [1] and Philippe Thomas [1]
[1] CRAN, Université de Lorraine, CNRS, Epinal, F-88000, France
E-mail: sylvain.chabanet@univ-lorraine.fr

Summary: In the sawmill industry, to predict the set of lumber that would be sawed from specific wood logs is a difficult problem. Even if they exist many sawmill simulators able to simulate the sawing process in order to predict these quantities, they can be too slow for large scale industrial problems. Replacing these simulators with machine learning surrogate models, or metamodels, is a promising avenue of research to speed up predictions. One such research direction is based on the computation of pairwise dissimilarities between logs, used, for example, by k-nearest neighbor algorithms. Interesting results have been obtained with the so-called iterative closest point (ICP) dissimilarity who has, however, several undesirable properties. This paper explores another alternative based on ensemble of shape functions.

Keywords: Sawmills, ICP dissimilarity, Ensemble of shape functions, k nearest neighbors, simulation metamodeling

1. Introduction

Sawmills are key elements of the forest product industry transforming wood logs into various lumber. Several factors, including heterogeneity of the raw material, introduce uncertainty on the mix of lumber that can be obtained from sawing a batch of logs. For this reason, academics and industrials have developed sawing simulators that are able to simulate the sawing of individual logs based on a description of their shapes, and sometimes internal defects. Shape information commonly comes in the form of 3D scans of the full profile of the logs, obtained using laser scanners (**Fig. 1**).

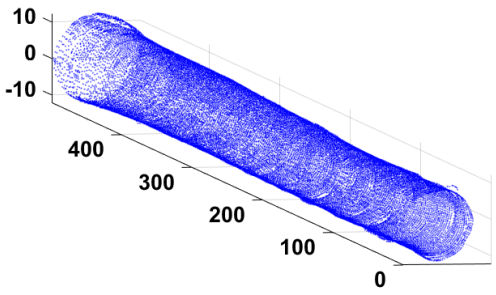

Fig. 1 Full profile 3D scan of a log. The scale is in centimeters.

These simulators can be used to support decision-making by alleviating the uncertainty associated with the sawing process, by predicting sets of lumber that might be obtained from every individual log [1]. In the following of this article, this set of lumber sawed from one log is called its basket of products (BoP).

However, the time taken by such simulation can be too long for practical use for decision problems involving thousands of logs. A single simulation can, indeed, take several minutes, or even more than one hour in some cases.

For this reason, researchers have proposed to replace these simulators with machine learning metamodels, i.e, surrogate models based on machine learning algorithms trained on past simulation results to predict BoP of new logs [2].

Few machine learning algorithms, however, allow making predictions based directly on 3D scans. These scans are, indeed, 3D points cloud, containing unordered points spanning the log surface. The number of points also varies from one scan to another.

Two main approaches have been proposed in past works to predict BoP of logs. The first one, introduced in [2], builds a structured representation of the logs based on a collection know-how features commonly used in the forest-product industry. The second approach, introduced in [3], is to compute a pairwise dissimilarity between 3D logs scans. These dissimilarities can then be used to predict BoP using, for example, a k nearest neighbors (kNN) algorithm.

Such a dissimilarity is a real-valued function $d(x_1, x_2)$, with x_1, x_2 two log scans, that intuitively measure how alike the two scans are. It is, in its usage, similar to a distance, but does not necessarily respect the properties of one, such as symmetry or positivity.

The dissimilarity used in previous works [3]–[5] to predict logs BoP is the so-called Iterative Closest Point (ICP) dissimilarity. This dissimilarity is a consequence of the Iterative Closest Point algorithm [6], which is classically used to align 3D point clouds. While this

dissimilarity led to interesting experimental results, it has several undesirable properties. Most importantly, it is not symmetric, and its results may depend on the point cloud orientations at the algorithm's initialization. This fact motivates the study proposed in this paper, which considers an alternative dissimilarity based on an ensemble of shape functions (ESF) evaluated on the 3D scans.

The remaining of this paper is structured as follows. Section 2 first introduces both the ICP and ESF-based dissimilarities, as well as their advantages. Numerical experiments comparing the performances of these dissimilarities to predict logs baskets of products are presented section 3. Section 4 concludes and proposes future research directions.

2. Dissimilarities computation

2.1. ICP dissimilarity

The ICP dissimilarity is a consequence of the Iterative Closest Point algorithm, which is an iterative algorithm for the fine registration of 3D shapes. This algorithm starts with two points clouds; one usually called the source and the other the target. It then searches for a rotation and a translation to minimize a position-dependent dissimilarity between the point clouds and align the source on the target.

The main steps of an iteration are as follows:

- Pair every point in the sources with its closest neighbor in the target. Points from the target may be selected several times or not at all. This step yield N_S pairs (s, t_s), with N_S the number of points in the source, s a point from the source and t_s its closest neighbor in the target.
- Find a rotation R and a translation T minimizing $D(R,T) = \sum_s (Rs + T - t_s)^2$. A closed-forms solution of this minimization problem can, in particular, be efficiently computed using quaternion theory [6].
- Apply the transformation obtained, go back to the first step, and loop until some ending criterium, such as a maximum number of iterations, is obtained.

It can be shown that the value of $D(R,T)$ decreases at every iteration of the algorithm. It, therefore, converges to a local minimum. The value of $D(R,T)$ obtained at the end of the last iteration of the algorithm is what is kept as the ICP dissimilarity d_{ICP}. Several inconveniences of this dissimilarity should, however, be noticed. Firstly, due to the non-symmetric roles of the target and source in the ICP algorithm, the ICP dissimilarity isn't symmetric, and in general $d_{ICP}(x_1, x_2)$ is not equal to $d_{ICP}(x_2, x_1)$. Secondly, the computation of d_{ICP} is highly dependent on the number of points in the source N_S and target N_T. More precisely, the complexity of one iteration of the ICP algorithm ranges from $O(N_S N_T)$ to $O(N_S \log(N_T))$ depending on the implementation of the closest neighbor search.

2.2. ESF dissimilarity

The ESF dissimilarity, d_{ESF}, is based on the representation of the logs scan as a collection of q normalized histograms $h_1, \dots, h_q$. These histograms have to be computed only once for every log, and can be stored and reused for multiple dissimilarities computation.

Every histogram h_j approximates the distributions of values taken by a shape function f_j evaluated over groups of points sampled at random from a scan. Various shape functions can be used. Three common functions were selected to be used in this paper. The first is the Euclidean distance between 2 points, the second is the angle defined by three points, and the third is the area of the triangle defined by three points. An example of these three histograms computed for one log is presented **Fig 2**.

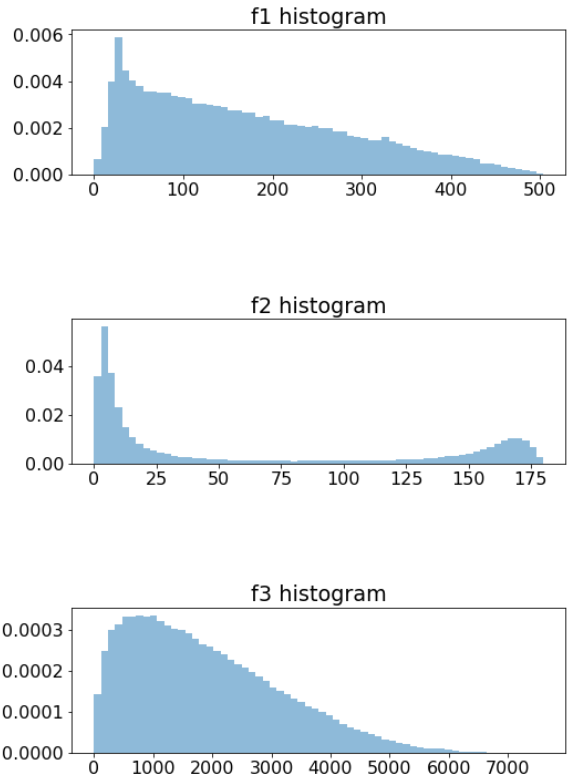

Fig 2. Histograms of the three shapes functions of a log scan used to compute the ESF dissimilarities.

The ESF dissimilarity between two logs is then defined as the sum of the L_2 distance between the histograms of both logs:

$$d_{ESF}(x_1, x_2) = \sum_{j=1}^{q} \| \mathrm{h}_{1\mathrm{j}} - h_{2j} \|_2, \qquad (1)$$

with h_{1j} and h_{2j} the histograms of the first and second log respectively.

Interestingly, the complexity of computing d_{ESF} does not depend on the number of points in the scans. It is, instead, governed by two parameters selected by the user. The first one is the number of pair or triplet of points selected at random to estimate the histogram representation of the scan. This parameter was set to 2^{17} ,in this paper, by trial and error in order to stabilize the histogram estimates. The second is the number of bins in the histograms, fixed to 64 here.

3. Experiments

This section presents numerical experiments comparing the performances of kNN algorithms based on either the ESF or ICP dissimilarity to predict BoP of logs. The dataset and evaluation scores are presented first, followed by the results.

3.1. Dataset

The dataset used for the experiments described in this paper originates from the Canadian sawmill industry. It contains information over 2219 real softwood logs. The 3D scan of every log is available, as well as their basket of products simulated by the sawing simulator optitek [7].

The scans of logs are composed of rough ellipsoids spanning the log surface. All scans are i initially oriented around the z-axis, with the first ellipsoid starting at z=0. This original orientation was kept as initial position when running the ICP algorithm to compute dissimilarities.

The sawmill modeled by the simulator could produce up to 74 types of lumber, characterized by their length, width, thickness, and grade (an evaluation of their quality). To simplify the prediction problem, the products were aggregated by grade, which reduce their number to 47, characterized only by their dimensions. A basket of products is, therefore, modeled as a vector of size 47. The i^{th} component represents the number of lumber of type I presents in the basket of products.

For experimental purpose, this dataset was repeatedly divided into a training set containing 1500 logs and a test set containing the remaining 719 logs. This dividing was repeated independently 30 times, the training set used as examples set by a kNN regressor algorithm, which is then evaluated on the test set.

Considering that the prediction problem is, here, modeled as a regression problem, the kNN prediction is the average of the baskets of the input log neighbors, and not necessarily a feasible basket. In particular, the kNN can predict non integer lumber quantities. This might not be a problem depending on the usage of the prediction, especially if, has done in [1], the predictions of individual logs are aggregated and used as input to a mix-integer programming problem.

3.2 Evaluation scores

Several evaluation scores are used in this study to evaluate and compare the predictive performances of the kNN algorithms using either dissimilarity.
The first one is the usual root mean squared error between real and predicted baskets:

$$RMSE = \sqrt{\frac{1}{N}\sum_{l=1}^{N}\sum_{i=1}^{47}(\hat{y}_{li} - y_{li})^2}\,, \qquad (2)$$

with N the size of the training set, $\hat{y}_{li}$ the predicted quantity of lumber of type i for the log l, and y_{li} the real quantity.

Several researchers, however, have stressed that such classic evaluation scores would be difficult to interpret for field experts from the industry and have proposed alternatives, in particular, the prediction-production score ($s^{pre\times pro}$) [2] and a variation of the F_1 score adapted to this problem [8].

In order to define the prediction-production scores, both the prediction score s^{pre} and production score s^{pro} need to be defined. Both are defined on a log-per-log basis.

The prediction score, s^{pre}, is the per-product average of the predicted lumber quantity over the real lumber quantity:

$$s_l^{pre} = \frac{1}{p}\sum_{i=1}^{p} \min\left(1, \frac{\hat{y}_{li}}{\max(y_{li}, \varepsilon)}\right). \qquad (3)$$

ε is, here, a very small quantity introduced to avoid dividing by 0. The index l in s_l^{pre} is added to stress the dependency over a specific log. Considering that this score is extremely sensible to $(0, 0)$ predicted-produced pairs which might make this score too optimistically biaised due to the sparsity of the produced and predicted basket of product, all such pairs are removed before computing this score. p is the number of non filtered products, which can vary from one log to another.

The production score, s^{pro}, is similarly defined as the per-product average of the real lumber quantity over the predicted lumber quantity:

$$s_l^{pro} = \frac{1}{p}\sum_{i=1}^{p} \min\left(1, \frac{y_{li}}{\max(\hat{y}_{li}, \varepsilon)}\right). \qquad (4)$$

The prediction-production score is then naturally defined as:

$$s^{pre\times pro} = \frac{1}{N}\sum_{l=1}^{N} s_l^{pre} \times s_l^{pro}. \qquad (5)$$

Similarly to the prediction-production scores, a variant of the F_1 score has been defined by [8] based on redefinitions of the numbers of True Positive (TP),

False Positive (FP) and False Negative (FN) computed on a log per log basis.

- The number of true positives TP_l is the number of lumber predicted and produced. $TP_l = \sum_{i=1}^{47} \min(\hat{y}_{li}, y_{li})$.
- The number of false positives FP_l is the number of lumber predicted but not produced. $FP_l = \sum_{i=1}^{47} \max(\hat{y}_{li} - y_{li}, 0)$.
- The number of false negatives FN_l is the number of lumber produced but not predicted. $FN_l = \sum_{i=1}^{47} \max(y_{li} - \hat{y}_{li}, 0)$.

The F_1 score is then redefined as:

$$s^{pre \times pro} = \frac{1}{N} \sum_{l=1}^{N} \frac{2 \times TP_l}{2 \times TP_l + FP_l + FN_l}. \quad (6)$$

3.3 Results

As detailed previously, the dataset was divided 30 times into a training and a test sets. For every dividing, two kNN algorithms searching neighbors with the ICP dissimilarity and ESF dissimilarity were trained on the train set and evaluated on the test set. In each case, the parameter k was tuned by 5 folds cross-validation on the training set, using the RMSE as basis for comparisons. k was selected among [1, 5, 10, 20]. All experiments were run on an Intel Core i7 vPRO 10th generation CPU at 2.70 GHz.

Experimental results are exposed **Table 1.** This table exposes the average and standard deviation over the repetitions of the experiments of the RMSE, prediction-production and F_1 scores for the kNN algorithms based on the ICP and on the ESF dissimilarities respectively.

The kNN based on the ESF dissimilarity has, in average, lower RMSE and higher prediction-production score and F_1 than the kNN based on the ICP dissimilarity. The poor performances of the ICP dissimilarity is, in part, due to the fact that it is not symmetric. In particular, replacing $d_{ICP}(x_1, x_2)$ by $d_{ICP}(x_1, x_2) + d_{ICP}(x_2, x_1)$ give a far lower RMSE, at 2.17. It, however, double the number of ICP computations needed to yield a prediction and is, therefore, not considered in the following.

To comfirm this difference between ICP and ESF dissimilarities, scores of both methods were compared by using student statistical test with Nadeau and Bengio correction [9]. This correction aim to take into account the dependency between results obtained for various dividing of the same dataset into a training and test sets. Given two prediction method A and B yielding evaluations a_j and b_j over J independent dividing of a dataset, the statistic of this test is:

$$t = \frac{\sum_j a_j - b_j}{\sqrt{(\frac{1}{J} + \frac{n_2}{n_1})\hat{\sigma}^2}}, \quad (7)$$

with n_1 the size of the training set, n_2 the size of the test set and $\hat{\sigma}^2$ an estimate of the variance of the differences $a_j - b_j$. The p-value of the test is then computed from the usual student distribution with J-1 degrees of freedom.

This test yield, here, a p-value of 3×10^{-3} when applied to the RMSE, 0.6 when applied to the prediction-production scores, and 5×10^{-15} when applied to the F_1. Therefore, the difference can be considered statistically significant for two of the three evaluation scores.

Table 1. Average and standard deviation of the evaluation scores of kNN regressors based on the ICP and ESF dissimilarities, taken over the 30 dividing of the datasets. The best model for each score is highlighted in bold.

Dissimilarity	RMSE	$s^{pre \times pro}$	F_1
ICP	3.05 (0.34)	34.8 (7.1)	35.3 (1.2)
ESF	**2.18 (0.25)**	**39.3 (8.0)**	**50.2 (1.0)**

The ESF dissimilarity might also appear interesting in terms of computational cost. The scan of logs used during the experiments exposed in this paper contains, in average, 18 452 points, and the average ICP dissimilarity computation time is 0.1 seconds, with 10 iterations of the algorithm. The implementation was based on Open3D library for python. The ESF dissimilarity was implement from scratch using the numpy python library. The computation of the collection of histograms for each logs took, in average, 5.8s per scan. These histograms, however, need to be computed only once for every log. In particular, when predicting the BoP of a log, only its own histograms need to be computed, because the others can be considered to have been computed and stored previously. Computing the ESF dissimilarity from pre-computed histograms is, then, extremely fast. In particular, it tooks only 0.0014s in average for the implementation used during experiments. Whether the ICP or ESF dissimilarities would be faster in practice would then depends on the specific user implementation, ability to parallelize the ICP computations and size of the kNN algorithm example set. The ESF dissimilarity appears, however, preferable for large example sets.

4. Conclusion

This paper explores an alternative to the ICP dissimilarity to predict BoP of logs based on their 3D scans. More precisely, it proposes the use of the ESF dissimilarity, based on the computation of an intermediary representation of the scans as an ensemble of histograms. The computation of this representation can take several seconds, but only needs to be computed once for every log. Computation of the

distance between the histograms is then far faster than computation of the ICP dissimilarity, which is advantageous for kNN algorithms with large example sets.

Additionally, when predicting BoP of logs with a kNN algorithm, the ESF dissimilarity leads to lower RMSE error and higher F_1 than the ICP dissimilarity.

Others machine learning algorithms, however, have been explored to predict BoP of logs from their 3D scans. These algorithms use a representation of a scan as a vector of dissimilarity toward a small set of preselected representative scans. Whether or not the ESF would still compare favorably for these algorithms needs to be explored in future works.

Acknowledgements

The authors gratefully acknowledge the financial support of the ANR-20-THIA-0010-01 Projet LOR-AI (lorraine intellgence artificielle) and région Grand EST.

We are also extremely grateful to FPInnovation who gathered and processed the dataset we are working with.

References

[1] M. Morin *et al.*, 'Machine learning-based models of sawmills for better wood allocation planning', *International Journal of Production Economics*, vol. 222, 2020, pp. 107508.

[2] M. Morin, F. Paradis, A. Rolland, J. Wery, J. Gaudreault, and F. Laviolette, 'Machine Learning-Based Metamodels for Sawing Simulation', in "*2015 Winter Simulation Conference (WSC)"*, 2015, pp. 2160-2171.

[3] C. Selma, H. B. El Haouzi, P. Thomas, J. Gaudreault, and M. Morin, 'An Iterative Closest Point Method for Measuring the Level of Similarity of 3D Log Scans in Wood Industry', in "*Service Orientation In Holonic and Multi-Agent Manufacturing*", 2018, pp. 433–444.

[4] S. Chabanet, P. Thomas, H. BRIL EL-HAOUZI, M. Morin, and J. Gaudreault, 'A kNN approach based on ICP metrics for 3D scans matching: an application to the sawing process', IFAC-PapersOnline, Vol. 54, Issue 1, 2021, pp. 396-401.

[5] S. Chabanet, V. Chazelle, P. Thomas, and H. B. El-Haouzi, 'Dissimilarity to Class Medoids as Features for 3D Point Cloud Classification', in *IFIP International Conference on Advances in Production Management Systems*, 2021, pp. 573–581.

[6] P. J. Besl and N. D. McKay, 'Method for registration of 3-D shapes', in *Sensor Fusion IV: Control Paradigms and Data Structures*, 1992, pp. 586-606.

[7] P. Goulet, 'Optitek - User's Manual'. 2006.

[8] V. Martineau, M. Morin, J. Gaudreault, P. Thomas, and H. B. El-Haouzi, 'Neural Network Architectures and Feature Extraction for Lumber Production Prediction', In *The 34th Canadian Conference on Artificial Intelligence*, Vancouvert, Canada, 25-28 May 2021, .

[9] R. R. Bouckaert and E. Frank, 'Evaluating the Replicability of Significance Tests for Comparing Learning Algorithms' In *Pacific-Asia conference on knowledge discovery and data mining*, 2004 pp. 3-12.