Machine Learning and Data in Sawing Simulation

Michael Morin Frédérik Paradis Amélie Rolland Jean Wery
Jonathan Gaudreault François Laviolette

UNIVERSITÉ LAVAL

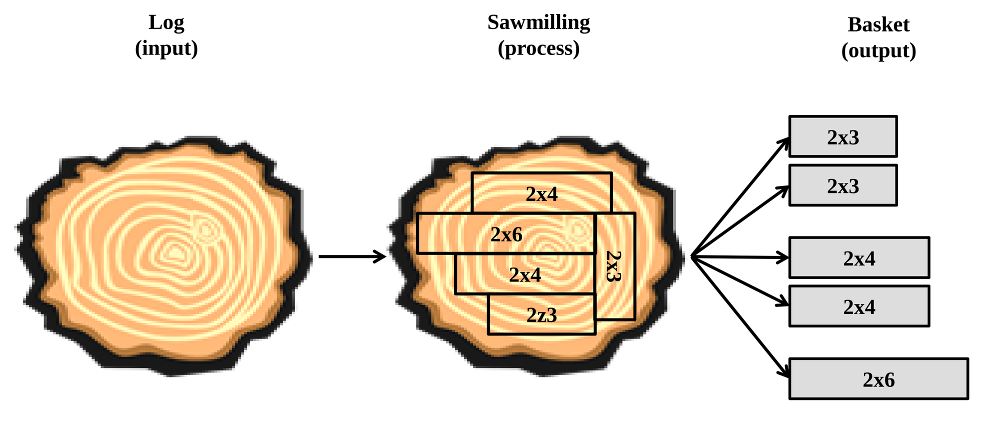
Simulations are used in sawmill modeling for decision-making. The inputs of a simulation are the data relative to the simulated sawmill and the three-dimensional representation of the available logs. During a simulation, the scanned logs are virtually transformed into a basket of lumbers according to a precise sawmill model. Decision-making involves various design scenarios and sets of virtual logs. Using these data, it is, for instance, possible for a decision maker to determine an efficient sawmill design. The approach is lengthy and the simulator requires a lot of fine-tuning.

We propose to use machine learning to predict the results of a log breakdown activity. While being faster than simulation, machine learning (metamodeling) also carries the advantage of allowing us to tackle complex variants of the sawing problem such as the converse problem of predicting the raw-matter needed to produce a given basket.

Sawing / Log Breakdown

From felled trees to lumbers:

1. Felled trees → logs → sawmill yard
2. The logs wait to be processed by the mill.
3. Single log → the basket (or mix) of products



A *divergent* process with *co-production*

Two different sawmill equipments process the exact same log differently. The *sawing pattern* of a log is constrained by:

- the *characteristics* of the log, e.g., its curvature;
- the *configuration of the sawmill*.

North American softwood *lumber products* are normalized according to National Lumber Grades Authority rules:

- the *type* of a product is defined by its thickness, its width, and its length;
- the *grade* of a product is a quality ordering relation among the products of the same type.

The sawing pattern influences the produced *basket* which is a set of normalized lumber products. The value of the lumbers varies in time. The sawmill equipment is configured to select a *feasible pattern* that maximizes the basket value.

Foreseeing the Outputted Lumbers is Capital

Strategic choices

- Long terms
- Sawmill design [1, 2, 3, 4, 5, 6]

Tactical decisions-making

- Medium terms
- Introduction of a new product [7]

Operational decision-making

- Short terms
- Which logs to process where and when? (logs inventory allocation)
- Which logs are valuable given the current trend? (lot buying and selling)

Simulation for Prediction

Given a sawmill model and its configuration, a sawmill simulator virtually transforms the logs.

Model (a *virtual sawmill*)

- Sawmill characteristics
- Equipment configurations

Input (a *virtual log*)

- The three-dimensional scan of a log; or
- the characteristics of a log:
  - curvature, narrow end and wide end diameter, length, volume, and shrinking.

Output (or *response*)

- the number of units of each feasible product

$$\begin{array}{cc} a\ b\ c\ d\ e & \textit{product} \\ [1\ 0\ 2\ 1\ 0] & \textit{count} \end{array}$$

Simulation is slow:

- for tactical and operational decision-making;
- for a large number of scenarios;
- for a large volume of wood.

Modeling takes time and building a simulator is non-trivial.

Supervised Machine Learning

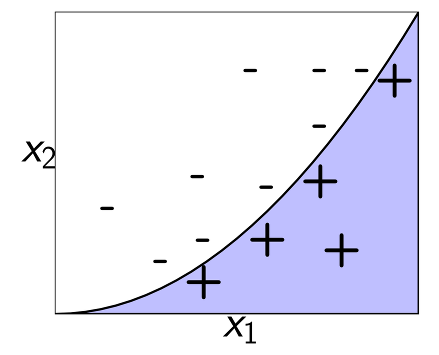
The learning algorithm has access to a training set of m examples

$$\mathcal{S} = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_m, \mathbf{y}_m)\}$$

where

- $\mathbf{x}_i = [x_{i1}, \ldots, x_{in}]$ are the inputs (features);
- $\mathbf{y}_i$ is the corresponding output, e.g., $+$ or $-$ for binary classification.

The goal of the learning algorithm is to approximate $h : \mathcal{X} \to \mathcal{Y}$.



The output of the learning algorithm is a clasifier $h : \mathcal{X} \to \mathcal{Y}$.

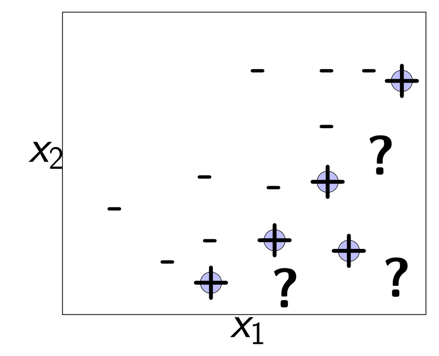
Performance Evaluation

The examples are assumed to be drawn iid (independently and identically distributed) from an unknown distribution D. The learning algorithm aims at finding h minimizing the expected loss (minimal prediction errors). Distribution D is, however, unknown. The expected loss is estimated on the training set $\mathcal{S}$:

$$r_{emp}(h) = \frac{1}{m} \sum_{i=1}^{m} l(h(\mathbf{x}_i), \mathbf{y}_i).$$

where $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is the incurred loss for an inaccurate prediction.

Minimizing empirical loss can lead to *over-fitting*.



Over-fitting leads to poor prediction performance.

Learning is a compromise between a classifier's performance and its complexity. The performance of a classifier h is assessed on unseen examples.

Variants of Supervised Learning

- Binary classification: classify into one of *two classes*.
- Multi-class: classify in *multiple classes*.
- Regression: output a *continuous value*.
- Structured output: output a *structure*.

Learning Algorithms

k-Nearest Neighbors (*k-NN*) [8]

Predicts the output by averaging the vote of the k closest examples in the training set.

Random Forest (*RF*) [9]

RF builds a forest of k decision trees where

- each *branching rule* partitions the examples;
- branching rules are learned in a greedy-fashion given a partitioning criterion;
- the value of a leaf is set to the majority class of the training examples reaching it.

The training phase is performed on k subsets of the training set leading to k trees. An unseen example is passed through each tree until a leaf is reached. A majority vote is performed.

Kernel Ridge Regression (*KRR*) [10]

KRR finds a linear operator $\mathbf{W}$ such that $h(\mathbf{x}) = \mathbf{W}\phi_{\mathcal{X}}(\mathbf{x}) \in \mathcal{H}_{\mathcal{Y}}$:

$$\begin{tikzcd} \mathcal{X} \arrow[r] \arrow[d, "\phi_{\mathcal{X}}"'] \arrow[dr, "\mathbf{W}\phi_{\mathcal{X}}(\mathbf{x})"] & \mathcal{Y} \arrow[d, "\phi_{\mathcal{Y}}"] \\ \mathcal{H}_{\mathcal{X}} \arrow[r] & \mathcal{H}_{\mathcal{Y}} \arrow[u, bend right, "\text{pre-image}"'] \end{tikzcd}$$

Metamodeling

Simulators are complex black box functions:

$$f\left(\left[\text{curvature, length}, \ldots\right]\right) = \begin{array}{cc} a\ b\ c\ d\ e & \textit{product} \\ [1\ 0\ 2\ 1\ 0] & \textit{count} \end{array}$$

Machine learning algorithms can be used to fit a metamodel

$$h : \mathcal{X} \to \mathcal{Y}$$

where $\mathcal{X}$ is the design space of the simulator and where $\mathcal{Y}$ is the response space of the simulator.

Once trained, the classifier h will be fast at predicting $\mathbf{y}$ from $\mathbf{x}$. MACHINE LEARNING can approximate a simulator's response. It can also approximate a REAL SAWMILL using the appropriate data.

Log Breakdown Prediction (LBP)

Given the (partial) characteristics of a log and the processing equipment (and/or plant), determine the resulting basket of products.

The training examples (data) needed:

- real or virtual logs characteristics;
- the basket resulting from the transformation of each log at a given plant.

The inputs needed for prediction purpose:

- the characteristics of an unseen log.

The output predicted:

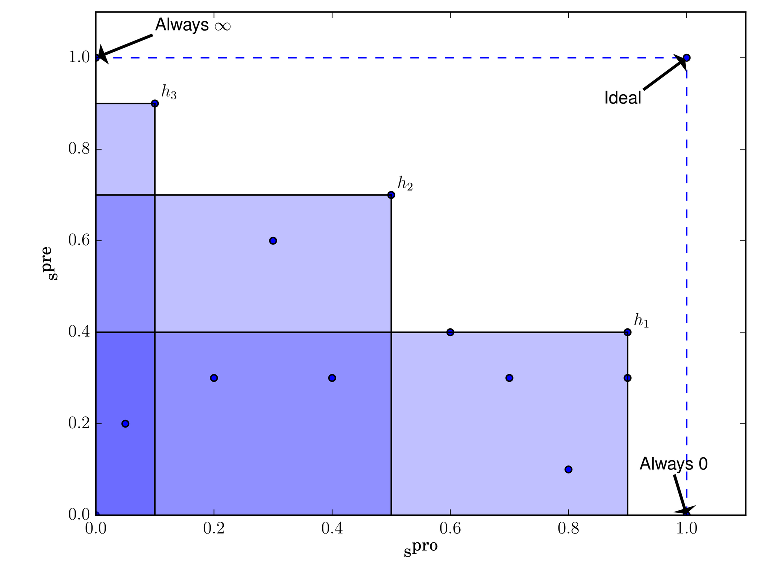
- for each product, the total number of lumbers of each type resulting from the transformation.

Experiments on Simulated Data

- 1207 baskets were generated from the three-dimensional scans of real logs using the Optitek simulator [11] along with an appropriate sawmill model.
- Each input vector contains the characteristics of a single log (curvature, narrow and wide end diameters, length, volume, and shrinking).
- The output is a vector containing the count of each of 19 plausible products considering the configuration of the equipments.

PRODUCTION RATIO (s^{pro}): The percentage of predicted products that were effectively produced.

PREDICTION RATIO (s^{pre}): The percentage of the real production that is faithfully predicted.



s^{pre} vs s^{pro}: Comparison of 11 fictive metamodels given $(\mathbf{x}, \mathbf{y})$

Results

Score	Average of 10 runs (standard deviation)			
	MEAN	*RF*	*KRR*	*k-NN*
Prediction ratio (s^{pre})	.6332 (.0077)	**.8945** (.0095)	.8885 (.0075)	.8635 (.006)
Production ratio (s^{pro})	.6096 (.0151)	.9099 (.0128)	.8946 (.0133)	**.9178** (.0134)

Faithfully predicted counts of 0 are removed to avoid boosting the scores of our classifiers.

Other Learning Opportunities

Some variants of the problem cannot be solved by simulation.

Raw-Matter Prediction

Given a basket and the processing equipment (and/or plant), determine the needed log characteristics.

References

[1] L. Orbay, "Computer simulation of overseas product manufacturing potentials of a redesigned multipass headrig mill in coastal British Columbia," 1984.

[2] D. Verret, *Modélisation du flux des bois dans les usines de sciage.* Thèse (M.Sc.), Université Laval, Québec, Canada, 1997.

[3] Q. J. Tong and S. Zhang, "Modelling jack pine lumber value recovery in relation to tree characteristics using Optitek simulation," *Forest Products Journal*, vol. 56, no. 1, pp. 66–72, 2006.

[4] C. G. Lundahl, *Optimized processes in sawmills.* PhD thesis, Luleå University of Technology, Sweden, 2007.

[5] C. Liu, S. Zhang, A. Cloutier, and T. Rycabel, "Modeling lumber value recovery in relation to selected tree characteristics in black spruce using the Optitek sawing simulator," *Forest Products Journal*, vol. 57, no. 4, pp. 57–65, 2007.

[6] P. Goulet, *Utilisation de la simulation du flux d'une scierie pour déterminer les gains en productivité associés au triage des tiges par dimensions.* Thèse (M.Sc.), Université Laval, Québec, Canada, 2007.

[7] J. Wery, P. Marier, J. Gaudreault, and A. Thomas, "Decision-making framework for tactical planning taking into account market opportunities (new products and new suppliers) in a co-production context," in *10ème Conférence Francophone de Modélisation, Optimisation et Simulation, MOSIM'14*, Nov. 2014.

[8] E. Fix and J. L. Hodges Jr, "Discriminatory analysis-nonparametric discrimination: consistency properties," tech. rep., DTIC Document, 1951.

[9] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[10] C. Cortes, M. Mohri, and J. Weston, "A general regression framework for learning string-to-string mappings," *Predicting Structured Data*, pp. 143–168, 2007.

[11] FPInnovations, "Optitek 10." User's manual, 2014.

http://www.MichaelMorin.info Michael.Morin.3@ulaval.ca