Achieving Accurate Printing Conditions Through Machine Learning

Torben Rapp and Philipp Tröster

Keywords: machine learning, artificial intelligence, printing condition, characterization, color management

Abstract

Printing conditions ensure that prints are produced consistently and to a high standard by describing all relevant parameters of the printing process. This is particularly important in industries where print quality is critical, such as in the production of packaging and labels.

In this paper, we present a novel approach to the classification of printing conditions using machine learning models. Our method involves the analysis of spectral measurement data to determine technical metadata such as the printing process, substrate type and print sequence. The accuracy of this data is crucial for setting up physical models, which can be used to predict overprints.

Previously, this data had to be manually entered which was error prone. However, our models can support the user by predicting this information accurately and quickly to ultimately ensure a high quality of prints.

We demonstrate the effectiveness of our method through automated and manual experimentation. Additionally, we describe the setup of an evaluation in a production environment.

Introduction

For predictable and reproducible prints, especially when it comes to high quality expectations in color, the printing behavior of a machine is often captured via a characterization. This includes creating and printing a test chart which is then measured. The set of parameters that describe under which conditions this characterization has been produced is then called the printing condition. Linking

GMG

the printing condition to the characterization and logging it accurately ensures that future prints yield the same result and color can be reproduced reliably.

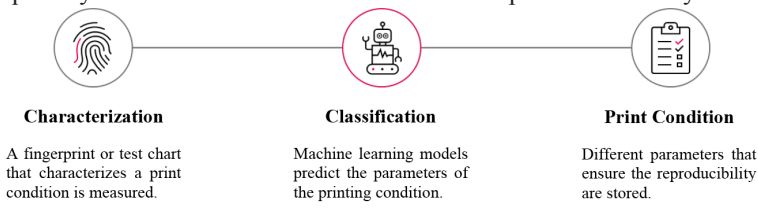

Figure 1. *For classifying a parameter for a printing condition, a machine learning model receives as input a characterization and predicts the corresponding parameter.*

However, for a given characterization, the corresponding printing condition under which the characterization has been produced, is not always known. This can occur for several reasons. Notably, it may happen because only the measured test chart is shared with third parties, or the original printing conditions were not permanently recorded. Classifying the printing condition correctly from a given test chart can be both time consuming and error prone. In this paper, we show that machine learning models can be used to support users by providing suggestions for printing conditions from a given characterization.

With this support, the time spent gathering information is reduced and correct parameters improve the next steps in the process, such as predicting overprints and searching in databases for similar, existing characterizations, ultimately resulting in higher quality color reproductions.

Methodology for predicting printing conditions from measurement data

The basic idea for predicting printing conditions can be seen in Figure 1 and is divided into three steps:

1. The model's input consists of spectral measurement data from a specific characterization, enriched with custom-engineered features.
2. The model itself classifies different parameters of the printing condition from the input data.
3. The printing condition is then the composition of the various parameters and can be used for further steps in the process.

The machine learning model can be a single model predicting the printing condition as a whole or as suggested in this paper, a composition of multiple models that focus on one single task – predicting individual parameters of the printing condition.

Parameter	Learning Task	Classes
Printing process	Classification	Flexo, Gravure, Offset, Digital
Front or reverse printing	Classification	Front, Reverse
Surface finishing	Classification	Laminate, Varnish, None
Substrate category	Classification	Paper, Cardboard, Film
Sequence of inks	Classification	$[ink_{io}, ..., ink_{ik}]^k$

Table 1. *Different parameters of a printing condition and their class representations.*

Data collection and analysis

Training a machine learning model using supervised learning requires known input data (characterizations) and known labels (printing condition parameters). The data that was used for this paper consists of roughly 100 measurement files with known printing conditions and was collected from printers and prepresses across different applications and using different printing conditions.

Five key parameters for printing conditions have been identified for having the biggest impact on the final color impression and one model was trained for each of them. See table 1 for details.

While one measurement file has exactly one class of each parameter assigned to it, the file itself consists of many patches: a combination of device values per ink used and the spectrum measured for it. The amount of data was not sufficient for training models on the entire measurement file, but with roughly 300,000 patches it was possible to train models patch-wise. Hence, we followed a two- step process for every parameter:

For every parameter:

1. Train one model per patch.
2. Assemble the predictions for each patch into one class.

The training itself is described in the section *Training the models*. The final predicted class is determined by selecting the most frequently predicted value per patch, except for the *sequence of inks*. The assembly for this parameter is explained in more detail in the aforementioned section.

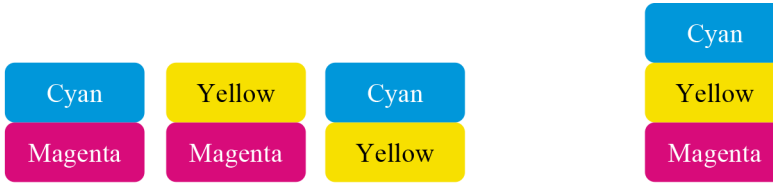

Figure 2. *Left: Pairwise prediction of which ink lies on top of the other. Right: Inks ordered based on the left pair-wise ordering. This represents the final ink sequence.*

Data preparation and feature engineering

For being able to train the models, all data must be brought into the same format. Since measurement files are basically tabular data consisting of rows (patches) and columns (device values and spectral values), the main challenge is to map all columns to the same schema. It is worth noting that measurement files mainly consist of two types of columns:

1. Columns representing the spectral values (e. g. “NM_380”)
2. Columns representing the device values (e. g. “CMYK_C”)

Additional columns include for example the CIE-Lab-values which can be ignored because they can be calculated from the spectral values.

Bringing the spectral columns for all (potential) measurement files into the same schema can be done easily by cropping to the range of 400—700 nm and reducing to a step distance of 10 nm in between which is the standard for most measurement devices anyways and sufficiently precise for the use case of predicting parameters.

The task of applying one schema to all potential inks and their device values is more challenging. For achieving a common mapping that also works for unknown inks outside the training data, we clustered the 100 % patches of all training data and map all inks into bins: 2 gray bins (light and dark gray) for inks with a small chromatic value and 6 chromatic bins for inks with a high chromatic value. The exact number of bins can be easily varied but proved to work well for the data we had available, including data for testing that was not part of the training data set.

Additionally, the quality of the predictions can be improved through feature engineering, i. e. adding more columns to the data set that precomputes some features making it easier for the models to learn correlations within a single class. The engineered features depend heavily on the available data. For example, we included the derivative of each spectrum as one feature. This way

the information on how big of a change lies within each range of the spectrum is more easily available to the models and improves overall quality.

Training the models

For every printing condition parameter, the training of a machine learning model is done in two steps:

For every printing condition parameter:

1. **Train the model**
   During training, the model learns relationships between the characterization (i. e. measurement data and additional features) and printing condition parameter classes (labels). 80 % of the available data is used for training.
2. **Evaluate the model**
   For evaluating the results, the remaining 20 % of the available data is used as test set to see with different metrics like accuracy and F1-score how well the model predicts the correct label for a given characterization.

The models used in this paper are gradient-boosting algorithms based on decision trees (Ke et al., 2017), which performed on our data significantly better than other algorithms tested. Useful models depend on various parameters, though, such as the underlying data and features.

Performance of different parameters

The quality of the predictions varies throughout the tasks. In table 2, the performances of the models on the various tasks are displayed.

It can be seen that e. g. for the printing process the accuracy of the predictions is very high. One reason for this is that the users who assign the printing condition to the characterization choose the correct printing process very precisely. Moreover, some processes have clearly distinguishable characteristics in the spectra, for instance the so called “flexo-bump”.

On the other hand, the accuracy for predicting the substrate category is not very high. It might be sufficient for an application but in one out of four cases it suggested the wrong substrate category to the user. One reason for this is that the categories in the training data were often incorrect. Hence, we manually labelled part of the data where we knew the correct category which ultimately resulted in a significantly smaller dataset.

	Test data		*Live data*	
Parameter	**Accuracy**	**F1-score**	**Accuracy**	**F1-score**
Printing process	0.917	0.938	0.803	0.783
Front or reverse printing	0.833	0.655	0.803	0.654
Surface finishing	0.930	0.870	0.906	0.239
Substrate category	0.857	0.681	0.721	0.488
Sequence of inks	0.833	0.500	0.819	0.574

Table 2. *Performance comparison of the trained machine learning models for different printing conditions parameters, evaluated per measurement file. The test data set consists of 20 % of the original characterization data set. The live data set, which continually expands through user interaction with the prototype, included 1,300 test charts at the time of publication.*

The most challenging parameter to predict is certainly the sequence of inks the measurement chart has been printed with. If we only did a pure classification on the possible combinations for inks, the number of possible classes would be higher than the number of characterizations we had which is infeasible for training. Instead, we broke down the task to a patch-wise prediction:

1. For all two ink overprints, predict which ink is printed on top of the other for every measurement chart.
2. Assemble the correct ink sequence from these pairs.

This improved the results overall but brought along new challenges: It can happen that for a 10 % Cyan, 10 % Yellow patch the model predicts that Yellow is printed on top and for the 20 % Cyan, 20 % Yellow patch the model predicts that Cyan is printed on top. These contradictions can be resolved via algorithms known from election theory. We used the well-known Schulze-method (Schulze,
2011) to address these inconsistencies. Nevertheless, it is not always possible to predict a print sequence at all, for example if there are two groups of inks in a measurement file without overprints between these two groups.

Evaluation in a production-like environment

Because the overall goal of this novel approach is to provide the user with suggestions to improve speed and quality of the entire process, we integrated this model into a production environment as a proof of concept. Table 2 compares the performance of this model on the test data set and in a live environment with real users across the industry.

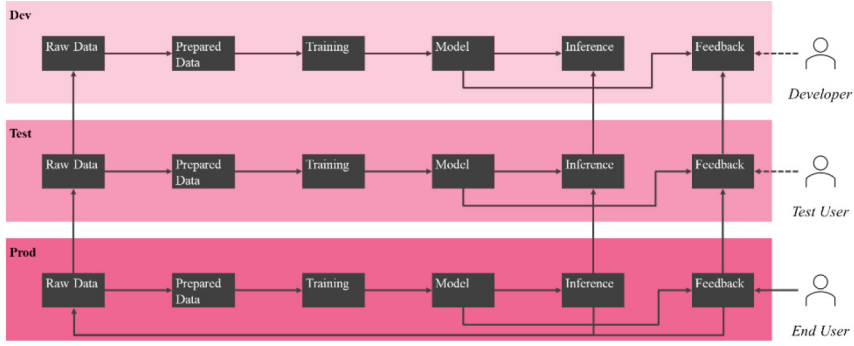

Figure 3. *Architectural overview of the three different environments Dev, Test, and Prod and the data flow between the individual modules. The raw data is being prepared for training and the final model is served for inference (making predictions for the user). The user gives feedback on the suggestions and the end user feedback is incorporated into new model investigation and training.*

Figure 3 shows the software architecture of how this model is integrated and uses the feedback from the user (accepts the suggestions or corrects them) for future training. It is also meant to provide a different angle on the "Hidden technical dept in machine learning systems" (Sculley et al., 2015).

Further research

The limited size of the data set has consequently restricted the number of models available for evaluation. With additional data, it would be interesting to explore models that operate on the entire measurement file, such as convolutional neural networks.

Moreover, the value for the end user should be investigated further. We introduced for instance a metric for the print sequence on how many inks a user must manually drag and draw to correct the suggested sequence of inks. This concept can be extended to the other parameters and give a realistic impression of "user satisfaction". Additionally, user surveys should be conducted to understand the impact for the end-user even better.

Conclusions

In this paper, we have demonstrated how machine learning models can be used to predict printing conditions from given measurement data. Specifically, we trained models to predict five different parameters of a printing condition. To evaluate the effectiveness of these models, we compared their performance on a

test data set and in a live user environment. Our results show that while the models performed slightly worse in the live setting compared to the test data set, their performance remained comparable and acceptable. This indicates that our machine learning approach is robust and capable of providing useful predictions in real-world applications, despite the inherent challenges and variability of live environments.

References

Ke, Guolin, et al. "Lightgbm: A highly efficient gradient boosting decision tree." *Advances in neural information processing systems* 30, 2017.

Schulze, Markus, "A new monotonic, clone-independent, reversal symmetric, and Condorcet-consistent single-winner election method", *Social Choice and Welfare*, volume 36, number 2, page 267–303, 2011. Preliminary version in Voting Matters, 17:9-19, 2003.

Sculley, David, et al. "Hidden technical debt in machine learning systems." *Advances in neural information processing systems* 28, 2015.