Dot Gain Table and Developer Voltage Prediction for the HP Indigo Press

Carl Staelin, Ruth Bergman, Mani Fischer, and Marie Vans
HP Laboratories Israel, Technion City, Haifa, ISRAEL

Darryl Greig
HP Laboratories Bristol, Stoke Gifford, Bristol, UNITED KINGDOM

Gregory Braverman, Shlomo Harush and Eyal Shelef
HP Indigo Division, Kiryat Weizmann, Rehovot, ISRAEL

Color consistency is crucial for both photo and commercial printing applications. Dot gain tables are updated regularly, however between updates colors can shift due to process drift in the press, which is a common problem of both digital and offset presses. The goal of this investigation is to dynamically control the dot gain table and developer voltage to ensure more consistent color control while minimizing waste and calibration measurements. In this article we approach the elements of this calibration process as a series of machine-learning problems and investigate the efficacy of replacing physical calibration measurements with model-based predictions. The current state of the machine, expressed as sensor measurements, is used to model both the developer voltage, and the subsequent dot gain look up table. We also consider models that make a prediction based on a restricted set of calibration measurements, not necessarily including the full machine state vector. Our initial investigation using a preliminary dataset shows that machine learning methods are suitable for predicting the dot gain table.

Journal of Imaging Science and Technology 49: 620–628 (2005)

Introduction

Color consistency is crucial for both photo and commercial printing applications. Look up tables (LUTs) for estimating dot gain values are currently updated on demand when the operator notices color consistency problems, and between updates colors can shift due to process drift in the press. The goal of the work presented in this article is to dynamically control the dot gain table and developer voltage to ensure more consistent color control while minimizing waste and calibration measurements.

Currently the dot gain table and developer voltage are controlled by printing special calibration test patterns on demand which are measured internally by the press. The calibration process begins by printing one or more test patterns with 100% ink coverage to determine a developer voltage setting for each ink such that the ink thickness at 100% coverage is within specification. Once the developer voltage is set, the actual ink thickness or optical density at 100% coverage is measured. Finally one or more sheets of test patterns with monochromatic swatches of uniform digital dot area are printed to measure the physical dot area for each of the digital dot areas. These measurements are the values required for the dot gain.

There are two distinct phases in this process which may be formulated as machine learning problems: (1) predict the developer voltage and corresponding ink optical density at 100% coverage per ink given the current machine state, and (2) predict the dot gain table values for each digital dot area of interest for each ink given the current machine state, developer voltage, and ink optical density at 100% coverage. A related problem, predicting the dot gain table given one or more measured dot values without any state information, is also examined here. A large number of machine learning regression algorithms are applicable to these problems. We evaluate the accuracy of three common methods: artificial Neural Networks (NN), Support Vector Machines (SVM), and linear regression. Neural networks are a well-known technique for machine learning. Both Bishop[1] and Ripley[2] give an excellent and readable treatment of theory and methods. Support vector machines are a kernel-based approach to machine learning. A good tutorial introduction to SVM was written by Burges.[3]

If a method is found to supply sufficiently accurate predictions, we can replace or augment the calibration procedure with a prediction-based process that has much less impact on customer workflow and consumable usage. The minimal requirements for the HP Indigo press are that the absolute difference between the predicted dot area and the printed dot area is less than 2 at least 67% of the time.

Original manuscript received February 1, 2005

Corresponding Author: Carl Staelin, Carl.staelin@hp.com

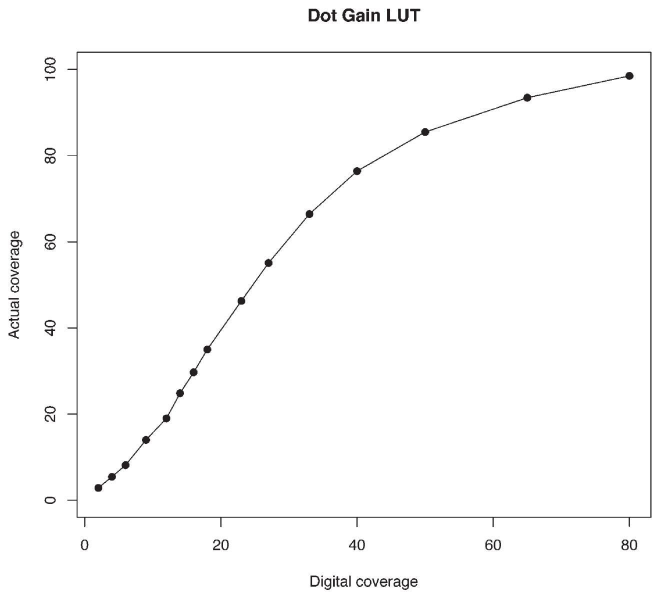

Figure 1. Sample dot gain curve. This curve was defined based on densitometer measurements of fifteen dot area values.

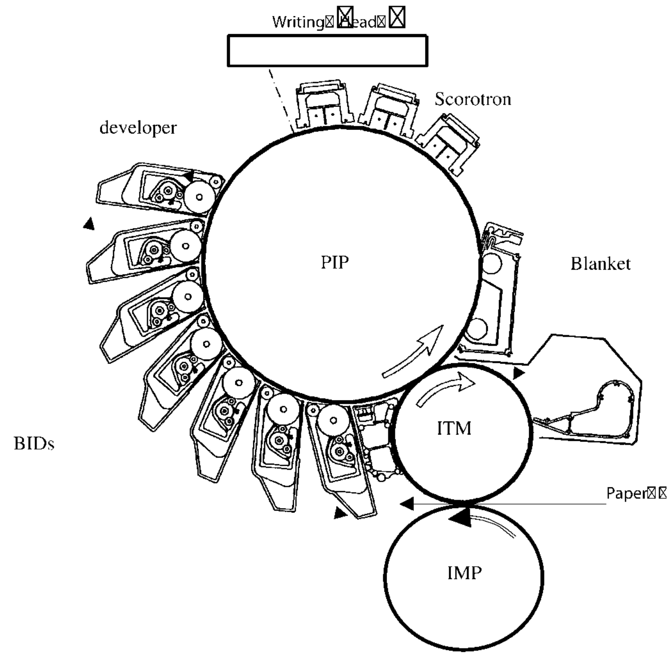

Figure 2. Indigo Press.

The dot gain is defined as follows:

$$dot\ gain = \frac{observed\ dot\ area}{digital\ dot\ area} \quad (1)$$

Both the digital dot area and observed dot areas are expressed as a percentage of the area that is covered, where 100 means that the whole area is covered with ink. The printed dot gain values are calculated using the densitometer measurements and the Murray–Davies equation.

Figure 1 shows an example dot gain curve. The dot gain curve is approximated from the physical dot area values of 15 digital dot area values by a piecewise linear function. The function is made up of 14 linear segments which connect between the known dot area values. While the dot gain curve is typically modeled from a smaller number of points, e.g., 4, by the gamma function or by two or three linear segments. In the case of the Indigo Press the curve usually has an S-shape which is not easily approximated by a parametric function nor by a small number of linear segments. Using fewer points affects the accuracy of the overall model. We will refer to the *dot gain lookup table* (LUT) which contains the printed dot area value, used in Eq. (1), for each of the fifteen points.

The calibration process uses an inline optical densitometer to read the printed dot areas from a swatch of uniform density in a single color. The current physical constraints of the system allow up to fifteen such swatches on a single sheet. Since the presses can have up to seven separations (inks), this implies that we may measure up to two digital dot areas for each separation in a single sheet.

The densitometer used for LUT calibration approximates the true dot area with a small measurement error. The practical accuracy of any model-based predictions is limited by the accuracy of the densitometer. In some cases, the models suggested here approach this accuracy limit.

As an alternative to the full calibration process (fifteen points), we consider a "fast calibration" process that measures one, two or more points per color separation, and then uses the measured information and the machine state to predict the rest of the dot gain lookup table values. This method does not change the underlying model used for the dot gain curve; the model remains a piecewise linear function with fourteen segments.

We analyzed a dataset of dot gain LUT's collected by HP Indigo. Our results for this dataset are promising in that the models give predictions within the required limits. It is important, however, to keep in mind that this dataset is small by machine learning standards – approximately 130 samples for each ink separation and halftone screen.

HP Indigo Dataset

The various parameters registered in the HP Indigo dataset are best understood by a brief introduction to the printing process of the HP Indigo Press.

The Printing Process

The process of image production consists of three stages (see Fig. 2). The first step is image generation in which a latent image is created on the Photo Imaging Plate (PIP) foil. The second step is image development. During this stage the latent image is developed by ink on the PIP. The last step is image transfer in which the developed image is transferred from the PIP to the Blanket that wraps the Intermediate drum (ITM). At this stage, the developed image is transferred from the Blanket to the substrate.

This process of converting a digital signal to a physical dot on a piece of paper can be affected by any number of system elements and interactions. Many key elements, such as the PIP foil and blanket are regularly replaced and each replacement part has its own characteristics.

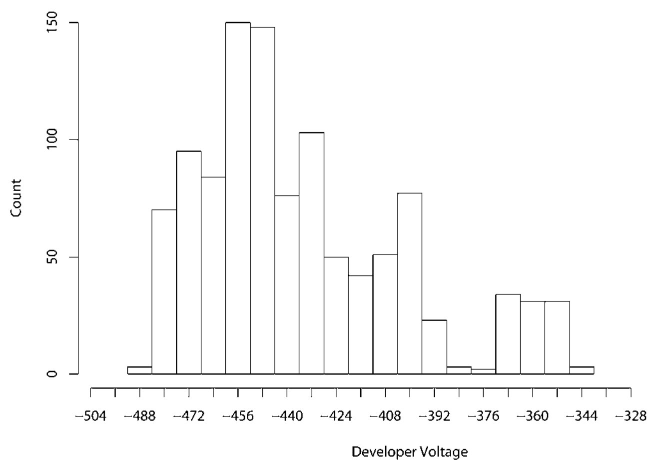

Figure 3. Histogram of Developer Voltage Values

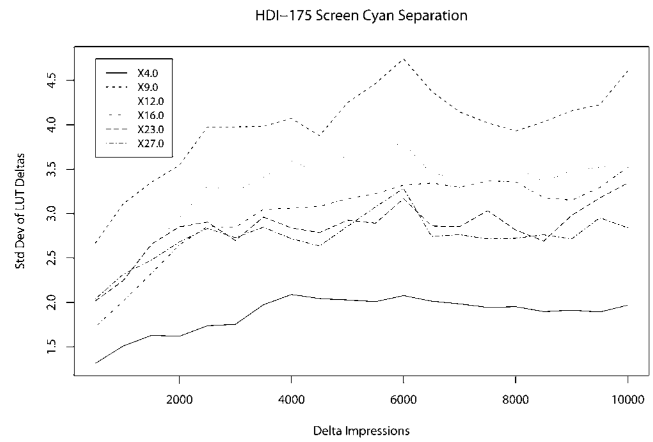

Figure 4. Standard Deviation of Printed Dot Area Differences for the Cyan separation.

Thus, it is likely that a full dot gain table measurement will need to be taken after each major part replacement. In addition, during normal operation, other parameters such as temperature, vary continuously.

Dataset Information

An experiment was conducted to collect the HP Indigo dataset from a single hp 1000 Press by a single operator. Note that Fig. 2 actually refers to the more sophisticated Series 2 press, but the essential printing process remains largely the same between the two models.

The experiment took place over a one week period. During this time the automatic dot gain calibration process was repeated every 1000 impressions (about 15 minutes). The dot gain LUT from every calibration was saved, i.e., fifteen printed dot area values for each of the four separations Black, Cyan, Magenta and Yellow. Associated with each dot gain LUT are a set of observed press parameters which are measured by sensors on the press. Twelve of these parameters are common across all separations, such as the ITM temperature, and seven others vary according to the current separation, e.g., ink characteristics and the developer voltage. The resulting dataset contains 269 dot gain tables in all. For a more detailed description of the dataset refer to Ref. 4.

All the graphs shown are for the 175 lpi HDI-175 screen. Similar results were obtained for the Sequin screen.

Developer Voltage Characteristics

The developer voltage for the HP Indigo Dataset is adjusted in steps of 8 V, although the final recorded voltage has some noise. A histogram of the total developer voltage observations separated into 8 V bins is given in Fig. 3. The distribution of the developer voltage values in this dataset appears to be at least bimodal, with the main mode occurring at –456, and a secondary mode at about –360. The extreme values in the bin centered on –488 and the bin centered on –344 are very under-represented, as are the bins –384 and –376, between the two modes.

Note that the adjustment step for developer voltage on Series 2 machines is significantly smaller that the 8 V step for the Series 1 machine represented in this dataset.

Dot Gain Variation Over Time

In order to deploy dot gain prediction effectively, we need to know how often the tables should be updated. If the tables are updated too frequently, then the color consistency can be reduced because the table values are changing faster than the underlying physical process. If the tables are updated too infrequently, then the press can drift out of control and color consistency is again reduced. The goal is to update the dot gain tables when it is likely that the press is drifting out of control.

As a first step we must determine how fast the press drifts as a function of the number of impressions. Viewing the dataset as a time series, we compared the measured dot gain values for each LUT with the same values in each subsequent LUT. The resulting data can be thought of as a function of LUT change versus the intervening impression count.

Figure 4 shows how the LUT table entries vary between measurements as a function of the number of intervening impressions. More precisely, it shows the standard deviation for changes in the LUT values. Since LUTs were not taken at fixed intervals, in terms of impressions, we binned the data into 500 impression buckets. Clearly the system can drift fairly quickly, so updating the LUTs as frequently as every thousand impressions would likely improve the color constancy.

One caveat with the results in Fig. 4 is that the developer voltage is adjusted as the first step in the calibration process, so between each measurement the develop voltage is updated. Thus this dataset is not representative of normal machine operation. During normal operation, the developer voltage would not be updated so frequently, so the actual variation in LUT values may be smaller than this graph indicates.

Developer Voltage Prediction

As noted in the introduction, the procedure for setting the developer voltage is an iterative procedure that requires significant consumable resources. Since our goal is to reduce both consumable waste, and machine "downtime", we wish to either replace or reduce the time and resources necessary for such procedures. In the case of developer voltage prediction, all of the measurable parameters are available to us with the exception of the optical density at 100% since this is measured as part of the developer voltage calibration procedure, and is thus dependent on the developer voltage.

The developer voltage observations were denoised before fitting models. That is, each observation was

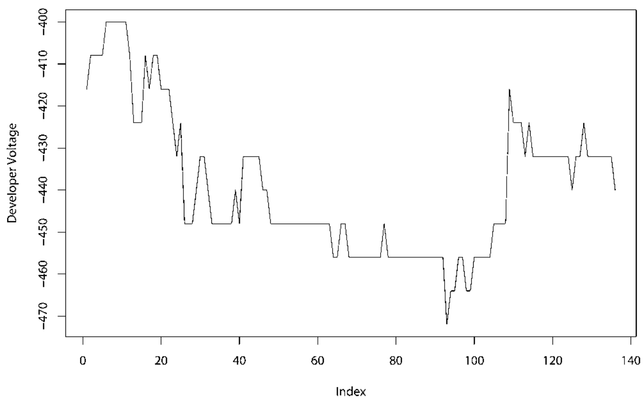

Figure 5. Developer Voltage for HDI-175 and Black Ink.

allocated the value of the nearest 8 V increment. This helps prevent the model from fitting noise artifacts.

Figure 5 shows an example of the developer voltage values for a single screen (HDI-175) and separation (Black) after this denoising operation.

The statistical learning problem associated with the prediction of developer voltage is an ordinal regression problem. In this problem formulation, a function of the predictors, i.e., the press parameters, is sought that predicts the rank of the developer voltage in the range of possible developer voltages. In the case of this data, there are nineteen classes, the lowest rank (class 1) being –488 V, and the highest rank (class 19) being –344 V. There are known linear and nonlinear techniques for solving this sort of problem (see, for example Ref. 5), however they generally require that all classes be well represented in the dataset. This is not the case with the dataset under consideration.

Therefore, we treat the problem as a simple regression problem, and then take the class nearest the model prediction as the predicted developer voltage. The fitted models interpolate in the under-represented regions and can therefore still provide predictions that should be sensible.

The data for developer voltage prediction, then, is the measured press parameters (excluding the optical density at 100% coverage). The developer voltage value is the desired output for the regression, and the remaining parameters are input. Prediction was evaluated in a ten-fold cross-validation experiment. Each combination of screen and ink comprised a separate experiment. We obtained developer voltage predictions for each of the statistical learning methods: linear regression, neural networks and support vector machines. The linear regression models included a stepwise parameter selection based on the AIC (see, for example, Ref. 6). The

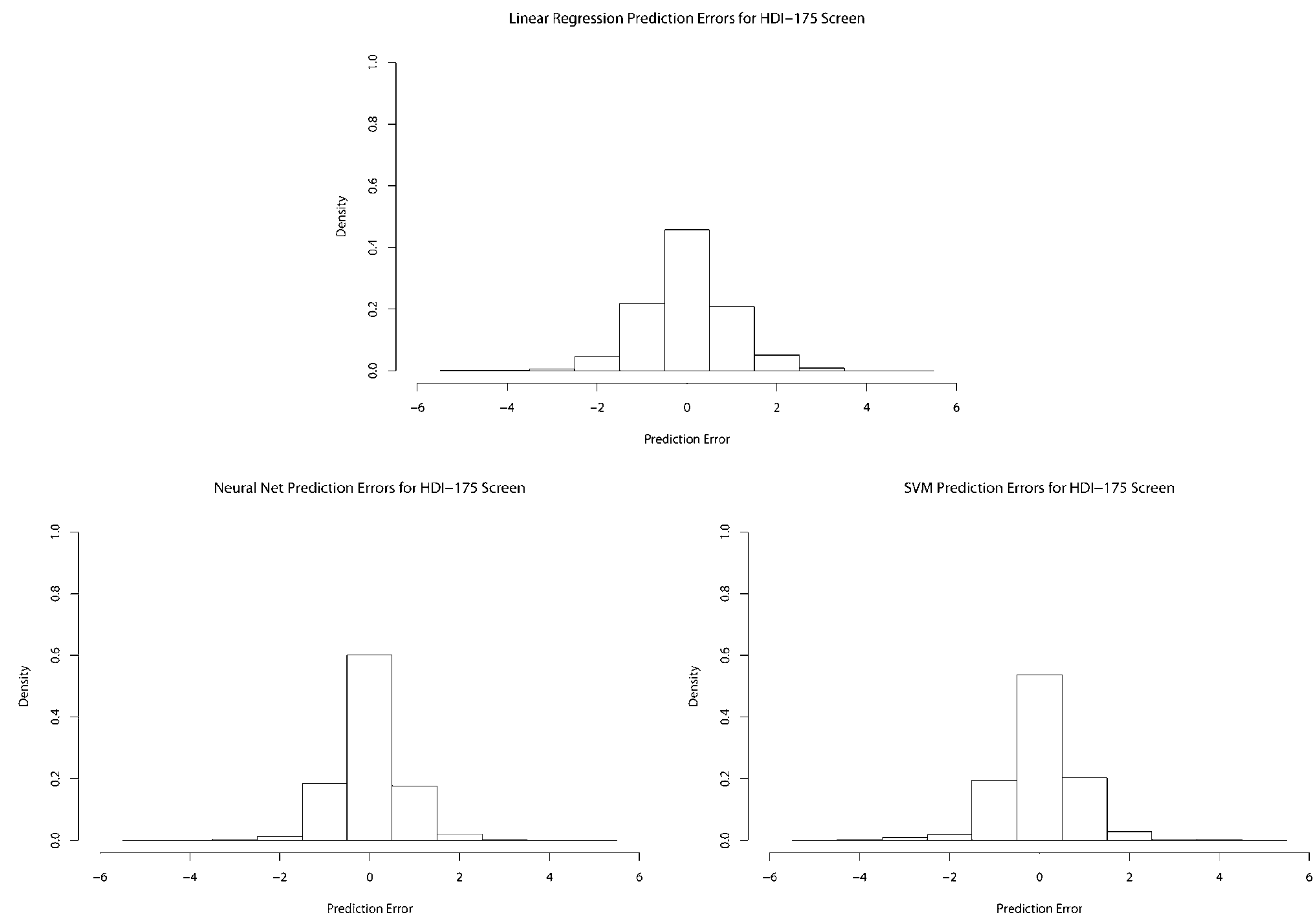

Figure 6. Prediction Error Histograms for Different Models.

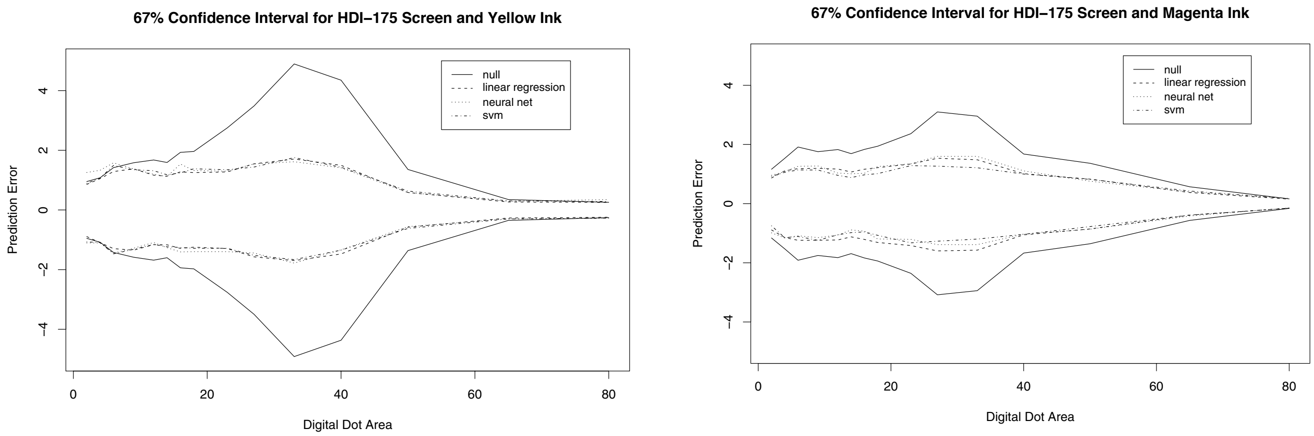

Figure 7. Prediction Error 67% Confidence Interval.

neural network models had a single hidden layer with 5 hidden nodes and a nonlinear output node. The support vector machine models used a radial basis function kernel, with hyper-parameters set by the parameter selection algorithm described by Staelin.[7]

As stated above, the resulting predictions are rounded to the nearest 8 V class. We then consider the discrepancy in number of classes between the predicted and actual values. That is, if a prediction of 432 is made when the actual class is 416, the reported error is on *predicted* – *true* = –2. The histograms of the resulting prediction errors for each statistical learning method are given in Fig. 6. In all cases the predicted voltage class is within 2 classes of the correct class for at least 99% of the predictions. For the linear regression models, the predicted class is within 1 class of the correct class at least 90% of the time, and both the nonlinear methods (neural networks and SVM) achieve a prediction within 1 class of the correct class at least 95% of the time, with neural networks slightly more accurate. The results are fairly consistent when broken down by separation.

Dot Gain Prediction Results

There are a variety of sub-problems under the general problem of dot gain LUT prediction that are appropriate for different usage models. The sub-problems we consider here fall into four general categories: (1) dot gain LUT predictions using only machine state measurements, i.e., no consumables are required; (2) parameter ranking/relevance for problem (1) models; (3) dot gain LUT predictions using the measurement of one or more dot gain values, i.e., without measuring any machine state variables.

For the prediction problems, the prediction errors are the difference between the true printed dot area and the predicted print dot area. The prediction errors were analyzed using a Chi-squared goodness of fit test and found that they are approximately normally distributed. Therefore, we can use the normal distribution multipliers for computing confidence intervals. The results are presented on graphs with the x axis giving digital dot area, and the y axis the difference of means. For a given model, a confidence "envelope" is plotted on this axis. That is, points corresponding to the upper limits of the confidence intervals for each digital dot area are joined to form a line, and the lower limits form a second line. This is simply for readability, since we often wish to compare multiple models on a single graph axis.

The graphs in Fig. 7 show the 67% prediction error confidence intervals for each of the three machine learning methods: linear regression, neural networks, and support vector machines as a function of the digital dot area. The null model prediction for each separation is also included for comparison. The Black and Cyan separations are shown. Note that each separation behaves slightly differently.

From the results in Fig. 7, it is apparent that the behavior of all the machine learning methods is similar. This means that the prediction of "hard" points is invariant of the learning method. Since linear regression performs comparably to the more complex nonlinear methods, all further analysis was done using linear regression. Note that the null model does not supply acceptable predictions.

Parameter Ranking and Selection

Some of the parameters used in the dot gain prediction models may be redundant. That is, we may find smaller models that fit the data equally well by removing some parameters. This has a two fold advantage – smaller models are more efficient computationally, and they are less prone to over-fitting. Furthermore, by analyzing the input parameters, we can gain some insight into the machine operation that may help the manufacturer identify other issues with parameter control etc.

The importance of a parameter may be measured by the effect of removing that parameter from the model. If the predictive power of the model is unaffected (or even improved) we may conclude that the parameter is not significant. On the other hand, if the model has a significant degradation in performance without a particular parameter, we may conclude that this parameter is significant and should be retained.

To implement this method we proceeded in the following manner. First model predictions were obtained for the entire dataset using the whole set of predictors. The predictions were obtained using the 10-fold cross validation technique.

The sum of squared errors (SSE) of these predictions was computed on each of the digital dot values of the LUTs, where both screens and all separations were included in this sum. Then similar predictions and SSE computations were made for models fitted *excluding* each of the input parameters in order.

We repeated this experiment 20 times and from this we were able to estimate the mean and standard

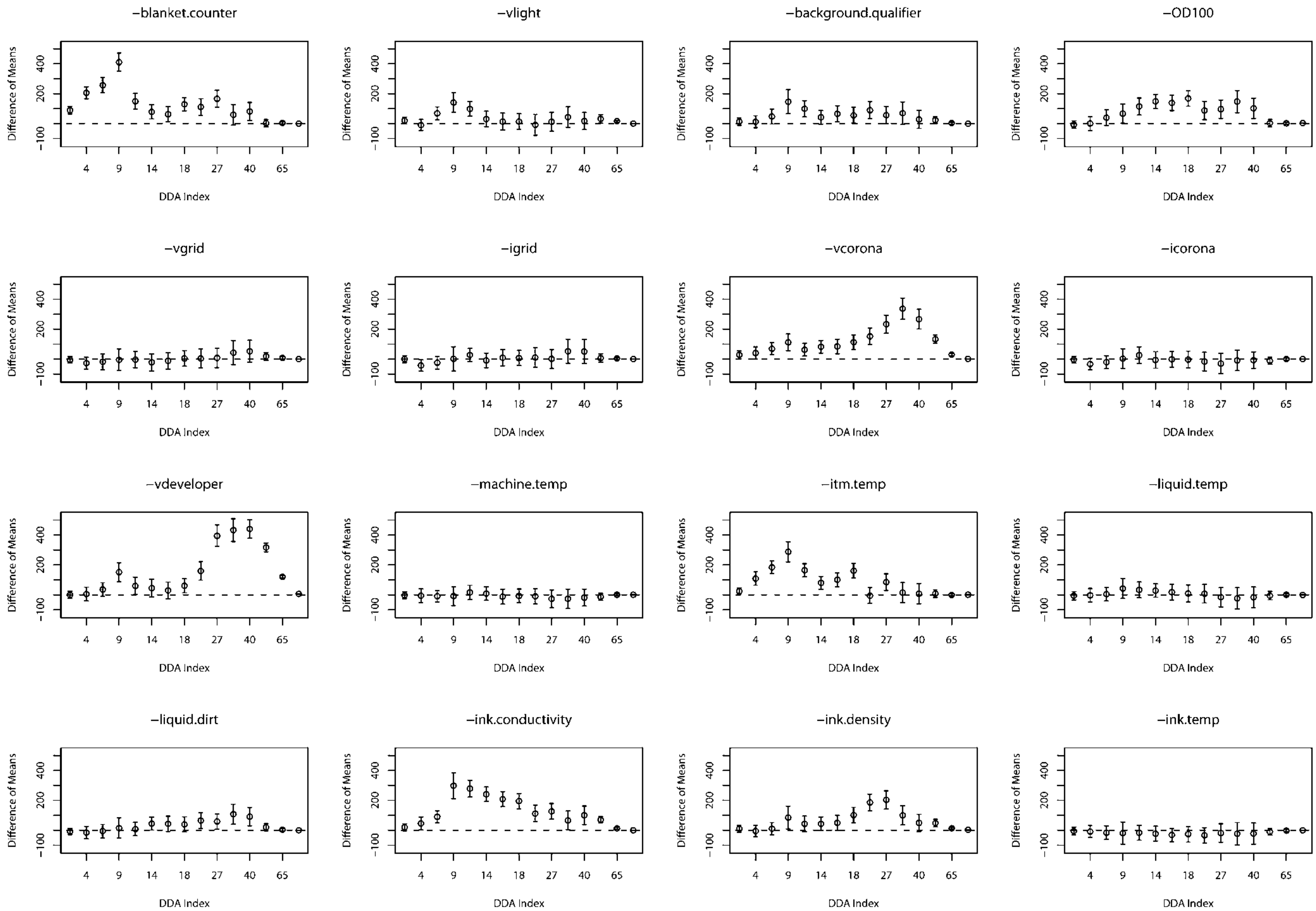

Figure 8. Difference of means 95% confidence intervals for depleted versus full linear regression models.

deviation of the SSE value for each of the table entries. Using these estimates we can generate confidence intervals for the differences of means between the original (full) model and each of the depleted models, for each DDA value.

Figure 8 summarizes these results. The error bars give the 95% confidence interval for the mean of the relevant depleted model SSE minus the mean of the full model SSE. If the confidence interval does not include the zero line in a particular case, then we can conclude (at the 0.05 level) that the parameter under consideration is relevant for the dot gain LUT prediction. From the graphs in Fig. 8, we can determine which parameters are significant to the prediction and which parameters are not significant. Note also that the significant parameters are effective at different DDA values. For example, the *blanket.counter* variable is effective mostly at the lower DDA values, while *vdeveloper* has the biggest effect at the high range of DDA values. We can also use this technique to obtain a ranking of the importance of each input parameter.[4]

Prediction with One or More Measured Points

The results in the previous section suggest that we can predict the dot gain LUT to within specification requirements by measuring the machine state. An alternative to eliminating the calibration process is reducing the waste due to calibration with a "fast calibration" where some digital dot area patches are printed and the printed dot area measured. In this section we quantify the prediction quality when using both one and two measured points.

This prediction problem immediately raises the issue of which points to add. From prior figures, such as Fig. 7 we can see that the error distribution tends to be bi-modal, with the smaller points, e.g., 2–16 covering one regime, and the middle points, e.g., 27–50 covering another regime.

Figure 9 shows the results of fitting models using a single measured LUT value, and predicting the remainder of the LUT based on this single value. For this case we used neural network models rather than linear regression models to add some nonlinearity to the predictions. The measured points in Fig. 9 are those that gave the best prediction results. These results suggest that such models can meet the accuracy requirement almost all the time.

Figure 10 shows the results of fitting models from two measured LUT values. Two sets of points are shown {27 & 50}, which gave the best single point prediction, and {16 & 40} which appear to give the best two point prediction. As expected, two point prediction gives better prediction results than one point and compares well to the prediction results using the machine state.

Cross-Screen Prediction

The machine operator may exchange a printing screen for a different screen during printer operation. Generally

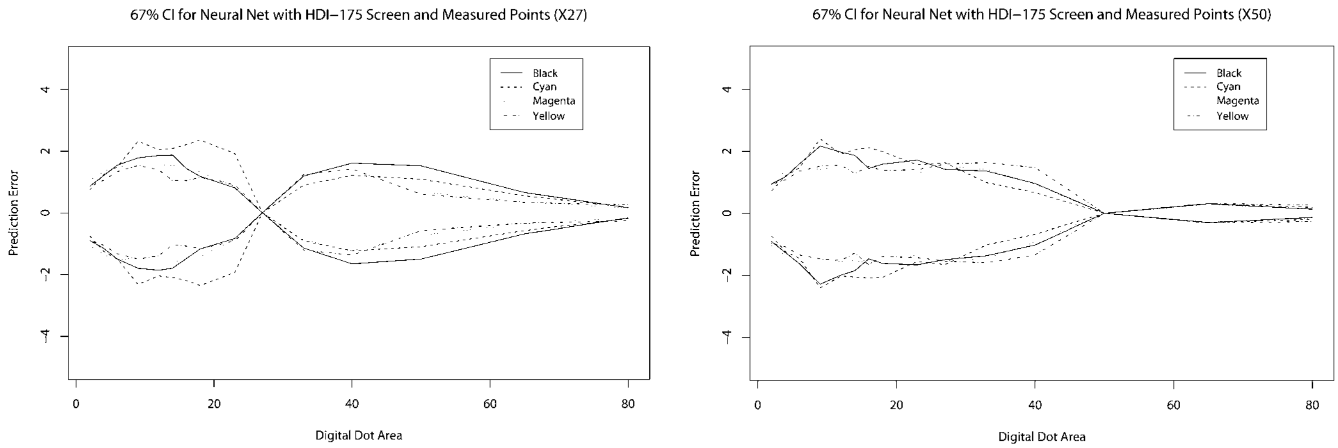

Figure 9. Prediction Error Confidence Intervals for Neural Network Predictions using One Measured Point and No Machine State.

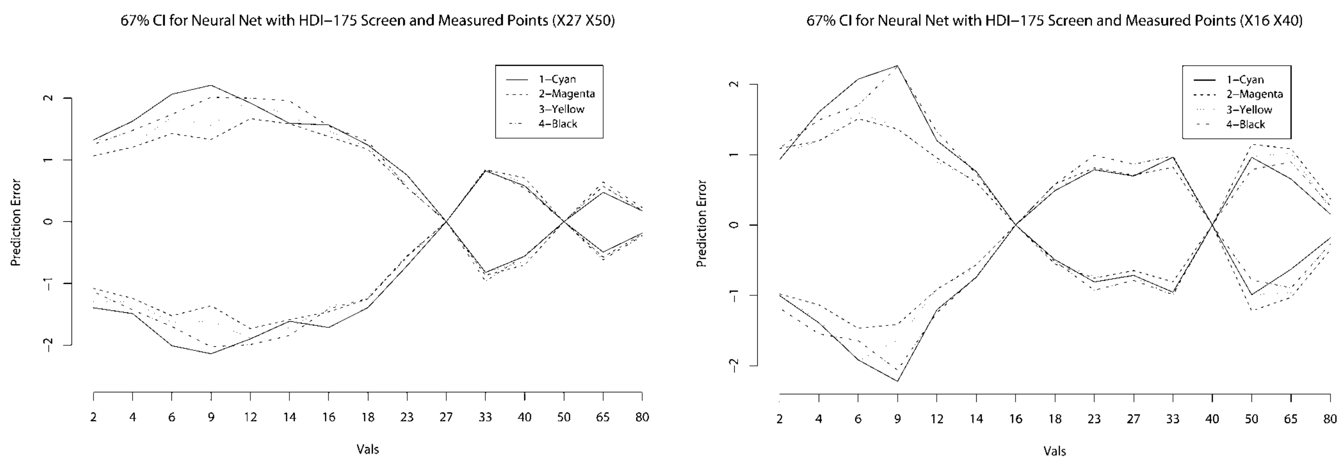

Figure 10. Prediction Error Confidence Intervals for Neural Network Predictions using Two Measured Points and No Machine State.

this process requires recalibration of various aspects of the machine, one of which is the dot gain LUT. If the machine state (temperatures, ink characteristics etc) does not change significantly during the screen exchange operation, then we may hope to discover a mapping between the dot gain LUTs for the two screens that will save some or all of the manual LUT calibration.

Using the existing dot gain LUT dataset, which contains roughly equal LUT values for the Sequin and HDI-175 screens, we were able to extract a subset of 84 paired LUT measurements, in which the LUT measurements for the two screens correspond to approximately the same machine state. Each measurement includes four separations, giving a total of 336 paired LUT samples. Since the goal here is to discover a mapping between a LUT of one screen type and a LUT of the other screen type, we treated all separations together. Figure 11 gives a plot of the printed dot area (PDA) values for one screen against the other, where each subplot corresponds to the digital dot area (DDA) given in the plot title. The apparent structure in these plots does suggest there is some relationship between the two screen LUTs, although it is quite weak in some cases.

In this investigation we shall only consider neural network models (two hidden nodes, skip layer connections, linear outputs, weight decay 0.001) and attempt to predict the PDA for each DDA on one screen based on the whole LUT for the corresponding other screen. Thus to predict the HDI-175 LUT (for example) from the Sequin LUT requires training seventeen networks on the learning dataset. The same number of networks is required to perform the reverse mapping. Since we are interested in finding a direct mapping between LUTs from two different screens we do not include any extra machine state parameters in the models; only the LUT values from the "source" screen are supplied as inputs.

The 67% confidence intervals for both comparisons (HDI→Sequin and Sequin→HDI) are show in Fig. 12. In both cases the predictions fall within an absolute difference of 2 at least 67% of the time. This meets Indigo's accuracy requirement for LUT prediction.

Conclusions

From the initial dataset it appears that given the measurable press parameters we can predict the various dot

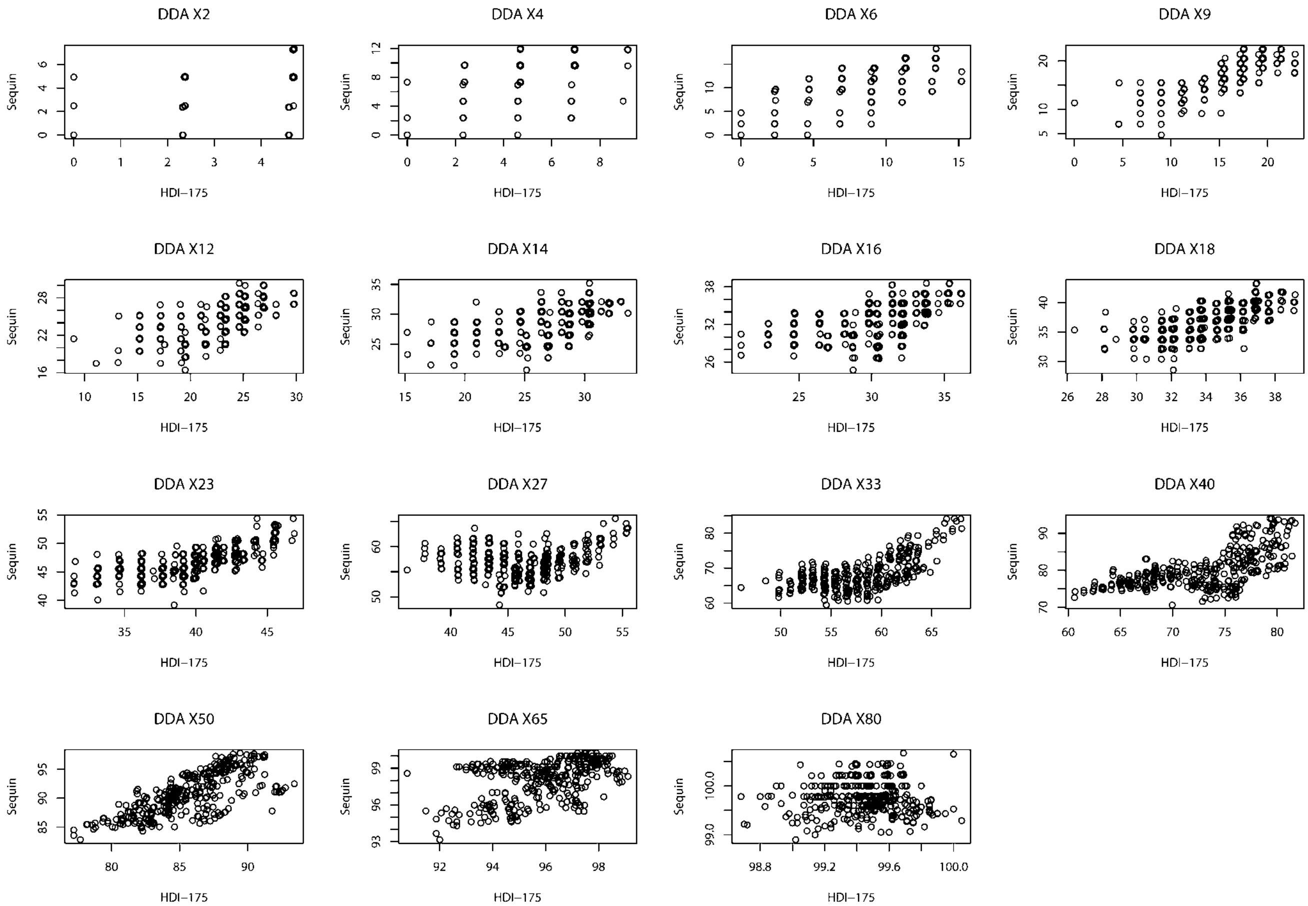

Figure 11. Scatterplots of PDA values from the Sequin Screen against PDA values from the HDI-175 Screen.

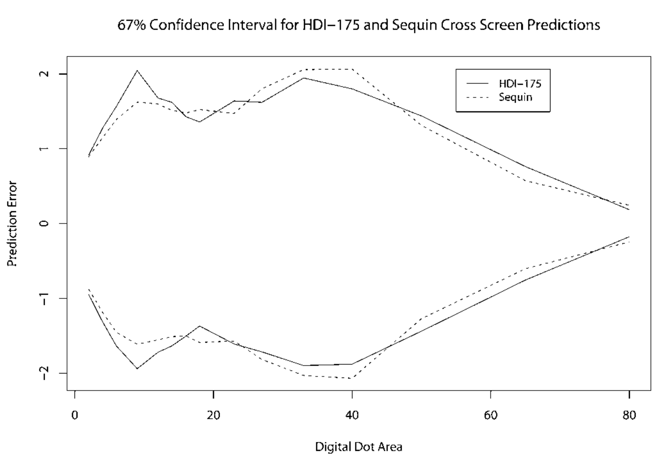

Figure 12. Prediction Error 67% Confidence Intervals for Cross-screen Prediction Using Neural Networks.

gain values with acceptable accuracy using linear regression. This should allow HP Indigo to greatly improve the color consistency for their presses, while reducing both the consumable waste and work-flow disruption.

We are surprised to see linear regression obtain results equivalent to the nonlinear learning methods (neural networks and SVM). It is possible that the relatively small dataset favors a simpler model, and that nonlinear models will perform better on a larger dataset. In future we plan to run more experiments using all three methods as we collect more data.

We suggested a method for assessing parameter importance. With this method we were able to conclude that some parameters do not significantly affect the performance of the model, and may, therefore, be eliminated. We also ranked the importance of the input parameters with respect to their effect on dot gain.

The results of dot gain LUT prediction based only on one or two measured points, suggest that there may also be opportunity to reduce consumable usage in some circumstances by printing a reduced LUT set "fast calibration", or relying on existing values. It does appear that there is a strong enough relationship between points in a dot gain LUT to be exploited by these simpler models.

This introductory study of the developer voltage prediction problem suggests that we are able to predict the developer voltage given the machine state parameters with a high degree of accuracy. In particular, if an error of at most one 8 V step is acceptable, then statistical learning methods can supply acceptable predictions more than 95% of the time. On the other hand, if an exact value is demanded, the models investigated here can give a starting point that will be accurate approximately 60% of the time, at most one step off approximately 95% of the time, and at most two steps off approximately 99% of the time. This may yield significant savings in consumables and calibration time for users of the Indigo press.

Once again, the developer voltage results are obtained on a relatively small dataset, operated with only two screens and a single substrate type. The conclusions obtained here should be verified on a much larger data sample, or alternatively as an experimental implementation on a functioning press. Future work, based on a larger dataset, may yield more accurate results as the more appropriate ordinal regression models could be utilized.

A number of questions need to be addressed before this can be sent to customers' presses. For example, we will need to evaluate the best update interval, e.g., how often should the system update the dot gain tables using model-based prediction with no printed measurements? How often should we use the "fast calibration" (one or more measured points) to get more accurate predictions? How often do we need to do full calibrations? How often should we update or refit the models to incorporate information from full calibrations? Other questions regard cross-machine measurement and prediction. For example, are individual presses idiosyncratic, or can we use measurements from one machine to predict the behavior of another machine?

References

1. C. M. Bishop, *Neural Networks for Pattern Recognition*, (Oxford University Press, Oxford, UK, 1995).
2. B. D. Ripley, *Pattern Recognition and Neural Networks*, (Cambridge University Press, Cambridge, UK, 1996).
3. C. J. C. Burges, "A tutorial on support vector machines for pattern recognition", *Data Mining and Knowledge Discovery*, **2**(2), 121, (1998). http://svm.first.gmd.de/papers/Burges98.ps.gz.
4. C. Staelin, R. Bergman, M. Fischer, D. Greig, M. Vans, G. Braverman, S. Harush, and E. Shelef, "Dot gain table prediction for the HP Indigo press", *Tech. Rep. HPL-2002-355*, (Hewlett-Packard Laboratories, Palo Alto, CA, 2002).
5. R. Bender and A. Benner, "Calculating ordinal regression models in SAS and S-Plus", *Biometrica*, **42**(6), 677 (2000).
6. R. Tibshirani T. Hastie and J. Friedman, *The Elements of Statistical Learning*, Springer Series in Statistics, (Springer, New York, NY, 2001).
7. C. Staelin, "Parameter selection for support vector machines", *Tech. Rep. HPL-2002-354,* (Hewlett-Packard Laboratories, Palo Alto, CA, 2002).