

Adaptive Local Linear Regression With Application to Printer Color Management

Maya R. Gupta, *Member, IEEE*, Eric K. Garcia, and Erika Chin

Abstract—Local learning methods, such as local linear regression and nearest neighbor classifiers, base estimates on nearby training samples, neighbors. Usually, the number of neighbors used in estimation is fixed to be a global “optimal” value, chosen by cross validation. This paper proposes adapting the number of neighbors used for estimation to the local geometry of the data, without need for cross validation. The term *enclosing neighborhood* is introduced to describe a set of neighbors whose convex hull contains the test point when possible. It is proven that enclosing neighborhoods yield bounded estimation variance under some assumptions. Three such enclosing neighborhood definitions are presented: *natural neighbors*, *natural neighbors inclusive*, and *enclosing k-NN*. The effectiveness of these neighborhood definitions with local linear regression is tested for estimating lookup tables for color management. Significant improvements in error metrics are shown, indicating that enclosing neighborhoods may be a promising adaptive neighborhood definition for other local learning tasks as well, depending on the density of training samples.

Index Terms—Color, color image processing, color management, convex hull, linear regression, natural neighbors, robust regression.

LOCAL learning, which includes nearest neighbor (NN) classifiers, linear interpolation, and local linear regression, has been shown to be an effective approach for many learning tasks [1]–[5], including color management [6]. Rather than fitting a complicated model to the entire set of observations, local learning fits a simple model to only a small subset of observations in a neighborhood local to each test point. An open issue in local learning is how to define an appropriate neighborhood to use for each test point. In this paper, we consider neighborhoods for local linear regression that automatically adapt to the geometry of the data, thus requiring no cross validation. The neighborhoods investigated, which we term *enclosing neighborhoods*, enclose a test point in the convex hull of the neighborhood when possible. We prove that if a test point is in the convex hull of the neighborhood, then the variance of the local linear

regression estimate is bounded by the variance of the measurement noise.

We apply our proposed adaptive local linear regression to printer color management. Color management refers to the task of controlling color reproduction across devices. Many commercial industries require accurate color, for example, for the production of catalogs and the reproduction of artwork. In addition, the rising ubiquity of cheap color printers and the growing sources of digital images has recently led to increased consumer demand for accurate color reproduction.

Given a CIELAB color that one would like to reproduce, the color management problem is to determine what RGB color one must send the printer to minimize the error between the desired CIELAB color and the CIELAB color that is actually printed. When applied to printers, color management poses a particularly challenging problem. The output of a printer is a nonlinear function that depends on a variety of nontrivial factors, including printer hardware, the halftoning method, the ink or toner, paper type, humidity, and temperature [6]–[8]. We take the *empirical characterization* approach: regression on sample printed color patches that characterize the printer.

Other researchers have shown that local linear regression is a useful regression method for printer color management, producing the smallest ΔE_{94}^* reproduction errors when compared to other regression techniques, including neural nets, polynomial regression, and tetrahedral inversion [6, Section 5.10.5.1]. In that previous work, the local linear regression was performed over neighborhoods of $k = 15$ NN, a heuristic known to produce good results [9].

This paper begins with a review of local linear regression in Section I. Then, neighborhoods for local learning are discussed in Section II, including our proposed adaptive neighborhood definitions. The color management problem and experimental setup are discussed in Section III and results are presented in Section IV. We consider the size of the different neighborhoods in Section V, both theoretically and experimentally. The paper concludes with a discussion about neighborhood definitions for learning.

I. LOCAL LINEAR REGRESSION

Linear regression is widely used in statistical estimation. The benefits of a linear model are its simplicity and ease of use, while its major drawback is its high model bias: if the underlying function is not well approximated by an affine function, then linear regression produces poor results. *Local* linear regression exploits the fact that, over a small enough subset of the domain, any sufficiently nice function *can* be well approximated by an affine function.

Manuscript received May 2, 2007; revised February 19, 2008. This work was supported in part by an Intel GEM fellowship, in part by the CRA-W DMP, and in part by the Office of Naval Research, Code 321, under Grant N00014-05-1-0843. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Reiner Eschbach.

M. R. Gupta and E. K. Garcia are with the Department of Electrical Engineering, University of Washington, Seattle, WA 98195 USA (e-mail: gupta@ee.washington.edu; eric.garcia@gmail.com).

E. Chin is with the Department of Electrical Engineering and Computer Science, University of California at Berkeley, Berkeley, CA 94720 USA (e-mail: erika.chin@gmail.com).

Digital Object Identifier 10.1109/TIP.2008.922429

Suppose that, for an unknown function $f : \mathbb{R}^d \rightarrow \mathbb{R}$, we are given a set of inputs $\mathcal{X} = \{x_1, \dots, x_N\}$, where $x_i \in \mathbb{R}^d$ and outputs $\mathcal{Y} = \{y_1, \dots, y_N\}$, where $y_i \in \mathbb{R}$. The goal is to estimate the output \hat{y} for an arbitrary test point $g \in \mathbb{R}^d$. To form this estimate, local linear regression fits the least-squares hyperplane to a local neighborhood \mathcal{J}_g of the test point, $\hat{y} = \hat{\beta}^T g + \hat{\beta}_0$, where

$$(\hat{\beta}, \hat{\beta}_0) = \arg \min_{(\beta, \beta_0)} \sum_{x_j \in \mathcal{J}_g} (y_j - \beta^T x_j - \beta_0)^2. \quad (1)$$

The number of neighbors in \mathcal{J}_g plays a significant role in the estimation result. Neighborhoods that include too many training points can result in regressions that oversmooth. Conversely, neighborhoods with too few points can result in regressions with incorrectly steep extrapolations. One approach to reducing the estimation variance incurred by small neighborhoods is to regularize the regression, for example by using ridge regression [5], [10]. Ridge regression forms a hyperplane fit as in (1), but the coefficients $\hat{\beta}$ instead minimize a penalized least-squares criteria that discourages fits with steep slopes. Explicitly

$$(\hat{\beta}^R, \hat{\beta}_0) = \arg \min_{(\beta, \beta_0)} \sum_{x_j \in \mathcal{J}_g} (y_j - \beta^T x_j - \beta_0)^2 + \lambda \beta^T \beta \quad (2)$$

where the parameter λ controls the trade-off between minimizing the error and penalizing the magnitude of the coefficients. Larger λ results in lower estimation variance, but higher estimation bias. Although we found no literature using regularized local linear regression for color management, its success for other applications motivated its inclusion in our experiments.

II. ENCLOSING NEIGHBORHOODS

For any local learning problem, the user must define what is to be considered local to a test point. Two standard methods each specify a fixed constant: either in the form of the number of neighbors k , or the bandwidth of a symmetric distance-decaying kernel. For kernels such as the Gaussian, the term “neighborhood” is not quite as appropriate, since all training samples receive some weight. However, a smaller bandwidth does correspond to a more compact weighting of *nearby* training samples. Commonly, the neighborhood size k or the kernel bandwidth is chosen by cross validation over training samples [5]. For many applications, including the printer color management problem considered in this paper, cross validation can be impractical. Consider that even if some data were set aside for cross validation, patches would have to be printed and measured for each possible value of k . This makes cross validation over more than a few specific values of k highly impractical. Instead, it will be useful to define a neighborhood that locally adapts to the data, without need for cross validation.

Prior work in adaptive neighborhoods for k-NN has largely focused on locally adjusting the distance metric [11]–[20]. The rationale behind these adaptive metrics is that many feature spaces are not isotropic and the discriminability provided by each feature dimension is not constant throughout the space. However, we do not consider such adaptive metric techniques appropriate for the color management problem because the

feature space is the CIELAB colorspace, which was painstakingly designed to be approximately perceptually uniform with three feature dimensions that are approximately perceptually orthogonal.

Other approaches to defining neighborhoods have been based on relationships between training points. In the *symmetric* k-NN rule, a neighborhood is defined by the test sample’s k-NN plus those training samples for which the test sample is a k-NN [21]. Zhang *et al.* [22] called for an “intelligent selection of instances” for local regression. They proposed a method called *k*-surrounding neighbor (*k*-SN) with the ideal of selecting a preset number k of training points that are close to the test point, but that are also “well-distributed” around the test point. Their *k*-SN algorithm selects NNs in pairs: first, the NN z not yet in the neighborhood is selected, then the next NN that is farther from z than it is from the test point is added to the neighborhood. Although this technique locally adapts to the spatial distribution of the training samples, it does not offer a method for adaptively choosing the neighborhood size k . Another spatially based approach uses the Gabriel neighbors of the test point as the neighborhood [23], [24, p. 90].

We present three neighborhood definitions that automatically specify \mathcal{J}_g based on the geometry of the training samples and show how these neighborhoods provide a robust estimate in the presence of noise. Because each of the three neighborhood definitions attempt to “enclose” the test point in the convex hull of the neighborhood, we introduce the term *enclosing neighborhood* to refer to such neighborhoods. Given a set of training points \mathcal{X} and test point g , a neighborhood \mathcal{J}_g is an enclosing neighborhood if and only if $g \in \mathbf{conv}(\mathcal{J}_g)$ when $g \in \mathbf{conv}(\mathcal{X})$. Here, the convex hull of a set $S = \{s_1, \dots, s_n\}$ with n elements is defined as $\mathbf{conv}(S) = \{\sum_{i=1}^n w_i s_i \mid w_i \geq 0, \sum_{i=1}^n w_i = 1\}$. The intuition behind regression on an enclosing neighborhood is that interpolation provides a more robust estimate than extrapolation. This intuition is formalized in the following theorem.

Theorem 1: Consider a test point $g \in \mathbb{R}^d$ and a neighborhood \mathcal{J}_g of n training points $\{x_1, \dots, x_n\}$ where $x_i \in \mathbb{R}^d$. Suppose g and each x_j are drawn independently and identically from a sufficiently nice distribution, such that all points are in general position with probability one. Let $f(g) = a^T g + a_0 + \omega_0$ and $f(x_j) = a^T x_j + a_0 + \omega_j$, where $a \in \mathbb{R}^d$, $a_0 \in \mathbb{R}$ and let each component of the additive noise vector $\omega = (\omega_0, \dots, \omega_n)^T \in \mathbb{R}^{(n+1)}$ be independent and identically distributed according to a distribution with finite mean and finite variance σ^2 . Given the measurements $\{f(x_1), \dots, f(x_n)\}$, consider the linear estimate $\hat{f}(g) = \hat{\beta}^T g + \hat{\beta}_0$, where $(\hat{\beta}, \hat{\beta}_0)$ solve

$$(\hat{\beta}, \hat{\beta}_0) = \arg \min_{\beta, \beta_0} \sum_{x_j \in \mathcal{J}_g} (f(x_j) - \beta^T x_j - \beta_0)^2. \quad (3)$$

Then, if $g \in \mathbf{conv}(\mathcal{J}_g)$, the estimation variance is bounded by σ^2

$$E_\omega \left[\left(\hat{f}(g) - E_\omega \left[\hat{f}(g) \right] \right)^2 \right] \leq \sigma^2.$$

The proof is given in Appendix A. Note that if $g \notin \mathbf{conv}(\mathcal{X})$ for training set \mathcal{X} , then the enclosing neighborhood \mathcal{J}_g cannot

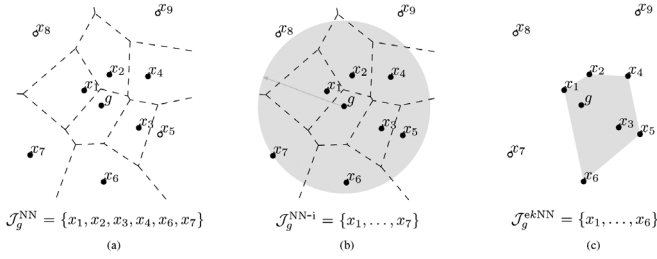


Fig. 1. (a) *Natural neighbors* neighborhood $\mathcal{J}_g^{\text{NN}}$ is marked with solid circles. For reference, the Voronoi diagram of this set is dashed. (b) *Natural neighbors inclusive* neighborhood $\mathcal{J}_g^{\text{NN-i}}$ is marked with solid circles; notice that $x_5 \in \mathcal{J}_g^{\text{NN-i}}$. The shaded area indicates the inclusion radius $\max_{x_j \in \mathcal{J}_g^{\text{NN}}} \{\|g - x_j\|_2\}$. (c) *Enclosing k-NN* neighborhood $\mathcal{J}_g^{\text{kNN}}$ is marked with solid circles.

satisfy $g \in \text{conv}(\mathcal{J}_g)$, and there is no bound on the estimation variance. In the limit of the number of training samples $n \rightarrow \infty$, $P\{g \in \text{conv}(\mathcal{X})\} = 1$ [3, Theorem 3, p. 776]. However, the curse of dimensionality dictates that for a training set \mathcal{X} with finite n elements and a test point g drawn *iid* in \mathbb{R}^d dimensions, the probability that $g \in \text{conv}(\mathcal{X})$ decreases as d increases [5], [25]. This suggests that enclosing neighborhoods are best-suited for regression when the number of training samples n is high relative to the number of feature dimensions d , such as in the color management problem.

Next, we describe three examples of enclosing neighborhoods.

A. Natural Neighbors

Natural neighbors are an example of an enclosing neighborhood [26], [27]. The natural neighbors are defined by the Voronoi tessellation \mathcal{V} of the training set and test point $\{\mathcal{X}, g\}$. Given \mathcal{V} , the natural neighbors of g are defined to be those training points x_j whose Voronoi cells are adjacent to the cell containing g . An example of the natural neighbors is shown in the left diagram of Fig. 1.

The *local coordinates property* of the natural neighbors [26] can be used to prove that the natural neighbors form an enclosing neighborhood when $g \in \text{conv}(\mathcal{X})$. Though commonly used for 3-D interpolation with a specific generalized linear interpolation formula called *natural neighbors interpolation*, Theorem 1 suggests that natural neighbors may be useful for local linear regression, as well. We were unable to find previous examples where the natural neighbors was used as a neighborhood definition for local regression or NN classification. One issue with natural neighbors for general learning tasks is the complexity of computing the Voronoi tessellation of n points in d dimensions is $O(n \log n)$ when $d < 3$ and $O((n/d)^{d/2})$ when $d \geq 3$ [28].

B. Natural Neighbors Inclusive

The natural neighbors may include a far training sample x_i , but exclude a nearer sample x_j . We propose a variant, *natural neighbors inclusive*, which consists of the natural neighbors and all training points within the distance to the furthest natural neighbor, $\max_{x_j \in \mathcal{J}_g} \|g - x_j\|_2$. That is, given the set of natural neighbors \mathcal{J}_g , the inclusive natural neighbors of g are

$$\left\{ x_j \in \mathcal{X} \mid \|g - x_j\|_2 \leq \max_{x_i \in \mathcal{J}_g} \|g - x_i\|_2 \right\}. \quad (4)$$

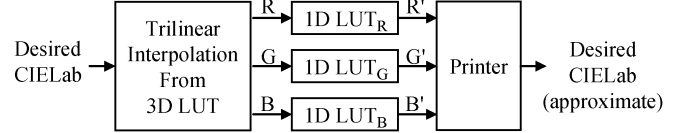


Fig. 2. Standard color management system: Desired CIELAB color is transformed to an appropriate RGB color that when input to a printer results in a printed patch with approximately the desired CIELAB color.

This is equivalent to choosing the smallest k -NN neighborhood that includes the natural neighbors.

An example of the natural neighbors inclusive neighborhood is shown in the middle diagram of Fig. 1.

C. Enclosing k-NN Neighborhood

The linear model in local linear regression may oversmooth if the neighbors are far from the test point g . To reduce this risk, we propose the neighborhood of the k -NNs with the smallest k such that $g \in \text{conv}(\mathcal{J}_g(k))$, where $\mathcal{J}_g(k)$ denotes the k -NNs of g . Note that no such k exists if $g \notin \text{conv}(\mathcal{X})$. Therefore, it is helpful to define the concept of a *distance to enclosure*. Given a test point g and a neighborhood \mathcal{J}_g , the distance to enclosure is

$$D(g, \mathcal{J}_g) = \min_{z \in \text{conv}(\mathcal{J}_g)} \|g - z\|_2. \quad (5)$$

Note that $D(g, \mathcal{J}_g) = 0$ if $g \in \text{conv}(\mathcal{J}_g)$.

Using this definition, the *enclosing k-NN neighborhood* of g is given by $\mathcal{J}_g(k^*)$ where

$$k^* = \min_k \{k \mid D(g, \mathcal{J}_g(k)) = D(g, \mathcal{X})\}.$$

If $g \in \text{conv}(\mathcal{X})$, this is the smallest k such that $g \in \text{conv}(\mathcal{J}_g(k))$, while if $g \notin \text{conv}(\mathcal{X})$ this is the smallest k such that g is as close as possible to the convex hull of $\mathcal{J}_g(k)$.

An example of an enclosing k -NN neighborhood is given in the right diagram of Fig. 1. An algorithm for computing the enclosing k -NN is given in Appendix B.

III. COLOR MANAGEMENT

Our implementation of printer color management follows the standard calibration and characterization approach [6, Section 5]. The architecture is divided into calibration and characterization tables in part to reduce the work needed to maintain the color reproduction accuracy, which may drift due to changes in the ink, substrate, temperature, etc. This empirical approach is based on measuring the way the device transforms device-dependent input colors (i.e., RGB) to printed device-independent colors (i.e., CIELAB). First, n color patches are printed and measured to form the training data \mathcal{X}, \mathcal{Y} , where $\mathcal{X} = \{x_i\}_{i=1:n}$ are the measured CIELAB values and $\mathcal{Y} = \{(y_{R_i}, y_{G_i}, y_{B_i})\}_{i=1:n}$ are the corresponding RGB values input to the printer. These training pairs are used to learn the LUTs that form the color management system. The final system is shown in Fig. 2: the 3-D LUT implements inverse device characterization which is followed by calibration by parallel 1-D LUTs that linearize each RGB channel independently.

A. Building the LUTs

The three 1-D LUTs enact gray-balance calibration, linearizing each RGB channel independently. This enforces that input neutral RGB color values with $R = G = B = d$ will print gray patches (as measured in CIELAB). That is, if one inputs the RGB colors (d, d, d) for $d \in \{0, \dots, 255\}$, the 1-D LUTs will output the RGB values that, when printed, correspond approximately to uniformly-spaced neutral gray steps in CIELAB space. Specifically, for a given neighborhood and regression method, the 918 sample Chromix RGB color chart is printed and measured to form the \mathcal{X}, \mathcal{Y} training pairs. Next, the L^* axis of the CIELAB space is sampled with 256 evenly-spaced values $\mathcal{G}_{L^*} = \{(100/255)(i - 1), 0, 0\}_{i=1:256}$ to form incremental shades of gray. For each $g \in \mathcal{G}_{L^*}$, a neighborhood \mathcal{J}_g is constructed and three regressions on $\{y_{R_i} | x_i \in \mathcal{J}_g\}$, $\{y_{G_i} | x_i \in \mathcal{J}_g\}$, and $\{y_{B_i} | x_i \in \mathcal{J}_g\}$ fit locally linear functions $h_c : \mathbb{R}^3 \rightarrow \mathbb{R}$ for $c = R, G, B$. Finally, the 1-D LUTs are constructed with the inputs $\{1, \dots, 256\}$ and outputs $\{h_c(g) | g \in \mathcal{G}_{L^*}\}$, where $c = R, G, B$ correspond to the three 1-D LUTs.

The effect of the 1-D LUTs on the training data must be taken into account before the 3-D LUT can be estimated. The training set is adjusted to find \mathcal{Y}' that, when input to the 1-D LUTs, reproduces the original \mathcal{Y} . These adjusted training sample pairs $\mathcal{X}, \mathcal{Y}'$ are then used to estimate the 3-D LUT (Note: In our process, all the LUTs are estimated from one printed test chart, as is done in many commercial ICC profile building services. More accurate results are possible by printing a second test chart once the 1-D LUTs have been estimated, where the second test chart is sent through the 1-D LUTs before being sent to the printer).

The 3-D LUT has regularly spaced gridpoints $g \in \mathcal{G}_{L^*a^*b^*}$. For the 3-D LUTs in our experiment, we used a $17 \times 17 \times 17$ grid that spans the CIELab color space with $L^* \in [0, 100]$ and $a^*, b^* \in [-100, 100]$. Previous studies have shown that a finer sampling than this does not yield a noticeable improvement in accuracy [6]. For each $g \in \mathcal{G}_{L^*a^*b^*}$, its neighborhood \mathcal{J}_g is determined, and regression on $\{y_{R'_i} | x_i \in \mathcal{J}_g\}$, $\{y_{G'_i} | x_i \in \mathcal{J}_g\}$, and $\{y_{B'_i} | x_i \in \mathcal{J}_g\}$ fits the locally linear functions $h_c : \mathbb{R}^3 \rightarrow \mathbb{R}$ for $c = R, G, B$.

Once estimated, the LUTs can be stored in an ICC profile. This is a standardized color management format, developed by the International Color Consortium (ICC). Input CIELAB colors that are not a gridpoint of the 3-D LUT are interpolated. The interpolation technique is not specified in the standard; our experiments used *trilinear interpolation* [29], a 3-D version of the common bilinear interpolation. This interpolation technique is computationally fast, and optimal in that it weights the neighboring grid points as evenly as possible while still solving the linear interpolation equations by choosing the maximum entropy solution to the linear interpolation equations [3, Theorem 2, p. 776].

B. Experimental Setup

The different regression methods were tested on three printers: an Epson Stylus Photo 2200 (ink jet) with Epson Matte Heavyweight Paper and Epson inks, an Epson Stylus

Photo R300 (ink jet) with Epson Matte Heavyweight Paper and third-party ink from Premium Imaging, and a Ricoh Aficio 1232C (laser engine) with generic laser copy paper. Color measurements of the printed patches were done with a Gretag-Macbeth Spectrolino spectrophotometer at a 2° observer angle with D50 illumination.

In our experiments, the calibration and characterization LUTs are estimated using local linear regression and local ridge regression over the enclosing neighborhood methods described in Section II and a baseline neighborhood of 15 NNs, which is a heuristic known to produce good results for this application [9]. All neighborhoods are computed by Euclidean distance in the CIELAB colorspace and the regression is made well posed by adding NNs if necessary to ensure a minimum of four neighbors. As analyzed in Section V, the enclosing k-NN neighborhood is expected to have roughly seven neighbors, where the word “roughly” is used to capture the fact that the assumptions of Theorem 2 (see Section V-A) do not hold in practice. The expected small size of enclosing k-NN neighborhoods led us to also implement a variation of the enclosing k-NN neighborhood which uses a minimum of $k = 15$ neighbors: this is achieved by adding NNs to the enclosing k-NN neighborhood if there are fewer than 15. Note that this variant is also an enclosing neighborhood, but ensures smoother regressions than the enclosing k-NN neighborhood.

The ridge parameter λ in (2) was fixed at $\lambda = 0.1$ for all the experiments. This parameter value was chosen based on a small preliminary experiment, which suggested that values of λ from $\lambda = .001$ to $\lambda = 2$ would produce similar results. Note that the effect of the regularization parameter is highly nonlinear, and that steeper slopes (higher values of β) are more strongly affected by the regularization. It is common wisdom that a small amount of regularization can be very helpful in reducing estimation variance, but larger amounts of regularization can cause unwanted bias, resulting in oversmoothing.

To compare the color management systems created by each neighborhood and regression method, 729 RGB test color values were drawn randomly and uniformly from the RGB colorspace, printed on each printer, and measured in CIELAB. These measured CIELAB values formed the test samples. This process guaranteed that the CIELAB test samples were in the gamut for each printer, but each printer had a slightly different set of CIELAB test samples. The test samples were then input as the “Desired CIELAB” values to test the accuracy of each estimated LUT, as shown in Fig. 2. Each estimated LUT produced estimated RGB values that, when sent to the printer, would ideally yield the test sample CIELAB values. The different estimated RGB values were sent to the printer, printed, measured in CIELAB, and the ΔE_{94}^* error was computed with respect to the test sample CIELAB values. The ΔE_{94}^* error metric is one standard way to measure color management error [6].

IV. RESULTS

Tables I–III show the average ΔE_{94}^* error and 95th percentile error for the three printers for each neighborhood definition, and each regression method. In addition, we discuss in this section which differences are statistically significantly different as judged at the .05 significance level by Student’s matched-pair

TABLE I
 ΔE_{94}^* ERRORS FROM THE RICOH AFICIO 1232C

Neighborhood	Regression	Average Error	95th Percentile Error
Enclosing k-NN	Linear	4.27	8.47
	Ridge	3.66	7.38
Enclosing k-NN Minimum 15	Linear	4.03	8.30
	Ridge	3.45	6.77
Natural Neighbors	Linear	3.74	7.55
	Ridge	3.69	7.10
Natural Neighbors Inclusive	Linear	3.74	7.63
	Ridge	4.03	7.70
15 Neighbors	Linear	4.41	9.84
	Ridge	4.16	8.61

TABLE II
 ΔE_{94}^* ERRORS FROM THE EPSON PHOTO STYLUS 2200

Neighborhood	Regression	Average Error	95th Percentile Error
Enclosing k-NN	Linear	2.32	5.01
	Ridge	2.20	5.03
Enclosing k-NN Minimum 15	Linear	2.34	5.00
	Ridge	2.32	4.95
Natural Neighbors	Linear	2.40	5.48
	Ridge	2.20	5.16
Natural Neighbors Inclusive	Linear	2.41	5.38
	Ridge	2.43	5.61
15 Neighbors	Linear	2.44	6.52
	Ridge	2.43	6.46

t-test. These three metrics (average, 95th percentile, and statistical significance) summarize different aspects of the results, and are complementary in that good performance with respect to one of the three metrics does not necessarily imply good performance with respect to the other two metrics. The baseline is the $k = 15$ neighbors with local linear regression. Small errors may not be noticeable; though noticeability varies throughout the color space and between people, errors under $2\Delta E_{94}^*$ are generally not noticeable.

The Ricoh laser printer is the least linear of the three printers, likely due to the printing instabilities that are common with high-speed laser printers. For the Ricoh, all of the enclosing neighborhoods have lower average error and lower 95th percentile error than the baseline of 15 neighbors and linear regression. Further, all of the methods were statistically significantly better than the 15 neighbors baseline, except for enclosing k-NN (linear), which was not statistically significantly different. Changing to ridge regression for 15 neighbors

eliminates over 10% of the 95th percentile error. Thus, the adaptive methods and the regularized regression make a clear difference for nonlinear color transformations.

For the Ricoh laser printer, the lowest average error and lowest 95th percentile error are produced by enclosing k-NN minimum 15 (ridge): the 95th percentile error is reduced by 21% over 15 neighbors (ridge), and the total error reduction is 31% over the baseline of 15 neighbors (linear). Enclosing k-NN minimum 15 (ridge) is statistically significantly better than all other methods for this printer. These results suggest that highly nonlinear color transforms can be effectively modeled by local regression using a lower bound on the number of neighbors (to keep estimation variance low) but allowing possibly more neighbors depending on their spatial distribution.

On the Epson 2200 all of the enclosing neighborhoods have lower average and 95th percentile error than the baseline of 15 neighbors (linear). However, only enclosing k-NN (ridge), enclosing k-NN minimum 15 (ridge), or natural neighbors (ridge)

TABLE III
 ΔE_{94}^* ERRORS FROM THE EPSON PHOTO STYLUS R300

Neighborhood	Regression	Average Error	95th Percentile Error
Enclosing k-NN	Linear	1.67	3.65
	Ridge	1.55	3.32
Enclosing k-NN Minimum 15	Linear	1.51	3.05
	Ridge	1.52	3.10
Natural Neighbors	Linear	1.71	3.49
	Ridge	1.54	2.87
Natural Neighbors Inclusive	Linear	1.77	3.55
	Ridge	1.79	3.55
15 Neighbors	Linear	1.55	3.32
	Ridge	1.55	3.34

were statistically significantly better (the other methods were not statistically significantly different). The natural neighbors (ridge) is statistically significantly better than all of the other methods except for enclosing k-NN (linear and ridge), which are not statistically significantly different. Enclosing k-NN (ridge) is statistically significantly better than all of the other methods except for natural neighbors (ridge). These results are consistent with the Ricoh results in that enclosing neighborhoods coupled with ridge regression provide significant benefit.

The Epson R300 inkjet fits the locally linear model well, as evident in the low errors across the board and the small average and 95th percentile error differences between methods. Here, few methods are statistically significantly different, but the natural neighbors inclusive is statistically significantly worse than the other neighborhood methods, including the baseline. We hypothesize that because the natural neighbors inclusive creates in some instances very large neighborhoods, this increase in error may be caused by the bias of oversmoothing.

We have presented and discussed our results in terms of the ΔE_{94}^* error because it is considered a more accurate error function for color management than ΔE_{76}^* (Euclidean distance in CIELAB) [6]. In (1) and (2), we minimize the Euclidean error in CIELAB, because this leads to a tractable objective, whereas minimizing ΔE_{94}^* error does not. The ΔE_{76}^* errors were also calculated and compared to the ΔE_{94}^* errors. The results were very similar in terms of the rankings of the regression methods and the statistically significant differences.

In summary, the experiments show that using an enclosing neighborhood is an effective alternative to using a fixed neighborhood size. In particular, enclosing k-NN minimum 15 (ridge) achieved the lowest average and 95th percentile error rates for the most nonlinear printer (the laser printer), and was either the best or a top performer throughout. Also, ridge regression showed consistent performance gains over linear regression, especially with smaller neighborhoods. Importantly, the overall low error rates on the inkjet printers suggest that the locally linear model fits sufficiently well on these printers, resulting in less room for improvement over the baseline method.

V. SIZES OF ENCLOSING NEIGHBORHOODS

Enclosing neighborhoods adapt the size of the neighborhood to the local spatial distribution of the training and test sample. In this section, we consider the key question, ‘‘How many neighbors are in the neighborhood?’’ We consider analytic and experimental answers to this question, and how the neighborhood size will relate to the estimation bias and variance.

A. Analytic Size of Neighborhoods

Asymptotically, the expected number of natural neighbors is equal to the expected number of edges of a Delaunay triangulation [27]. A common stochastic spatial model for analyzing Delaunay triangulations is the Poisson point process, which assumes that points are drawn randomly and uniformly such that the average density λ is n points per volume S . Given the Poisson point process model, the expected number of natural neighbors is known for low dimensions: six neighbors for two dimensions, $48\pi^2/35 + 2 \approx 15.5$ neighbors for three dimensions, and $340/9 \approx 37.7$ neighbors for four dimensions [27].

The following theorem establishes the expected number of neighbors in the enclosing k-NN neighborhood if the training samples are sampled from a uniform distribution over a hypersphere about the test sample.

Theorem 2 (Asymptotic Size of Enclosing k-NN): Suppose n training samples are uniformly sampled from a distribution that is symmetric around a test sample in \mathbb{R}^d . Then, asymptotically as $n \rightarrow \infty$, the expected number of neighbors in the enclosing k-NN neighborhood is $2d + 1$.

The proof is given in Appendix C.

For both the natural neighbors and enclosing k-NN, these analytic results model the training samples as symmetrically distributed about the test point. This is a good model for the general asymptotic case where the number of training samples $n \rightarrow \infty$, because if the true distribution of the training samples is smooth then the random sampling of training samples local to the test sample will appear as though drawn from a uniform distribution.

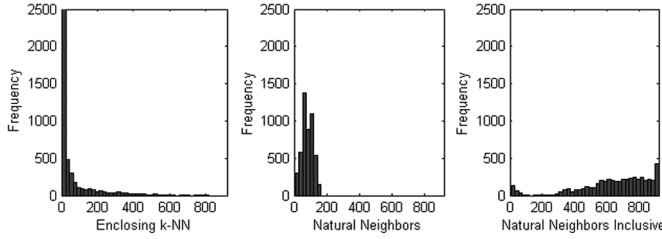


Fig. 3. Histograms show the frequency of each neighborhood size when estimating the gridpoints of the 3-D LUT for the Ricoh Aficio 1232C.

B. Experimental Size of Neighborhoods

The analytic neighborhood size results suggest that the natural neighbors is a larger neighborhood on average than the enclosing k-NN neighborhood, which we found to be true experimentally. Representative empirical histograms of the neighborhood sizes are shown in Fig. 3. They show the distribution of the neighborhood sizes for the color management of the Ricoh printer from 918 training samples.

By design, the enclosing k-NN is the smallest possible k-NN neighborhood that encloses the test point, which should keep estimation bias relatively low because the neighbors are relatively local. The natural neighbors tend to form larger neighborhoods than enclosing k-NN, and a particular natural neighbor could be close or far from the test sample. Thus, it is hard to judge how local the natural neighbors are. The natural neighbors inclusive has relatively large neighborhoods, which suggests that some of the natural neighbors must in fact be quite far from the test sample. The large size of the natural neighbors means that the estimated transforms will be fairly linear across the entire colorspace, which can oversmooth the estimation.

One cause of large neighborhood sizes is when a test point is outside the convex hull of the entire training set. As discussed in Section I, the set of training samples may not span the full colorspace, resulting in exactly this situation. An illustration of how such cases affect the enclosing neighborhood sizes is provided in Fig. 4. Here, the enclosing k-NN neighborhood is x_1 . From the Voronoi diagram, one can read that the natural neighbors of g are $\{x_1, x_2, x_3, x_4, x_7\}$. The largest of the neighborhoods in this case is the natural neighbors inclusive, composed of $\{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$.

When training and test samples are drawn *iid* in high-dimensional feature spaces, the test samples tend to be on the boundary of the training set, an effect known as Bellman's *curse of dimensionality* [5], [25]. We hypothesize that this effect for high-dimensional feature spaces would cause an abundance of large neighborhoods for the natural neighbors methods.

The inclusion of possibly far-away points to "enclose" the test point may result in increased bias. Based on the neighborhood size histograms and our analysis of the different neighborhoods, enclosing k-NN should incur the lowest bias. On the other hand, we expect natural neighbors inclusive to have the largest positive effect on variance because a larger neighborhood tends to lead to lower estimation variance for regression problems, though this matter is not so straightforward for classification.

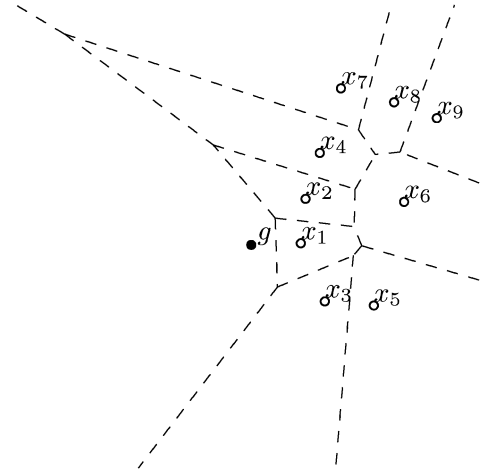


Fig. 4. Voronoi diagram of a situation where the test point g lies outside the convex hull of the training samples.

VI. DISCUSSION

We have proposed the idea of using an *enclosing neighborhood* for local learning and theoretically motivated it for local linear regression. Such automatically adaptively-sized neighborhoods can be useful in applications where it is difficult to cross validate a neighborhood size, and in particular we have shown that using enclosing neighborhoods can significantly reduce color management errors. Local learning can have less bias than more global estimation methods, but local estimates can be high-variance [5]. Enclosing neighborhoods limit the estimation variance when the underlying function does have a (noisy) linear trend. Ridge regression is another approach to controlling estimation variance, but does so by penalizing the regression coefficients, which increases bias. In contrast, we hypothesize that the effect of an enclosing neighborhood on bias may be either positive or negative, depending on the actual geometry of the data. It remains an open question how the estimation bias and variance differ between enclosing neighborhoods and the standard k-NN, which uses a fixed, but cross validated, k for all test samples.

The definition of the neighborhood for local learning is important, whether for local regression, or for local classification. We conjecture that using enclosing neighborhoods for other local learning tasks may lead to improved performance, particularly for densely sampled feature spaces.

APPENDIX A PROOF OF THEOREM 1

Proof: Form the $(d + 1)$ -dimensional vectors $\tilde{g} = [g \ 1]^T$ and $\tilde{a} = [a \ a_0]^T$. Let $m = |\mathcal{J}_g|$, and re-index the training samples so that $x_i \in \mathcal{J}_g$ for $i = 1, \dots, m$. Let $\tilde{\mathbf{X}}$ denote the $(d + 1) \times m$ matrix whose i th column is $[x_i \ 1]^T$. Further, let $\tilde{\beta} = [\hat{\beta} \ \hat{\beta}_0]^T$, let y be the $m \times 1$ vector with i th component $f(x_i)$, and let $\tilde{\omega}$ be the $m \times 1$ vector with i th component ω_i . The least-squares regression coefficients which solve (3) are

$$\begin{aligned} \tilde{\beta} &= (\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T)^{-1}\tilde{\mathbf{X}}y \\ &= (\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T)^{-1}\tilde{\mathbf{X}}(\tilde{\mathbf{X}}^T\tilde{a} + \tilde{\omega}) \\ &= \tilde{a} + (\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T)^{-1}\tilde{\mathbf{X}}\tilde{\omega}. \end{aligned}$$

Note that $E[\tilde{\beta}] = \tilde{a} + (\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T)^{-1}\tilde{\mathbf{X}}E[\tilde{\omega}]$. Let I denote the $m \times m$ identity matrix. Then the covariance matrix of the regression coefficients is

$$\begin{aligned} \text{cov}(\tilde{\beta}) &= E \left[\left(\tilde{\beta} - \tilde{a} - (\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T)^{-1}\tilde{\mathbf{X}}E[\tilde{\omega}] \right) \right. \\ &\quad \left. \times \left(\tilde{\beta} - \tilde{a} - (\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T)^{-1}\tilde{\mathbf{X}}E[\tilde{\omega}] \right)^T \right] \\ &= (\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T)^{-1}\tilde{\mathbf{X}}\text{cov}(\tilde{\omega})\tilde{\mathbf{X}}^T(\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T)^{-1} \\ &= (\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T)^{-1}\tilde{\mathbf{X}}\sigma^2 I \tilde{\mathbf{X}}^T(\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T)^{-1} \\ &= \sigma^2(\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T)^{-1}. \end{aligned}$$

The variance of the estimate $\hat{y} = \tilde{\beta}^T \tilde{g}$ is

$$\begin{aligned} \text{var}(\hat{y}) &= E \left[(\hat{y} - E[\hat{y}])^2 \right] \\ &= E \left[\left(\tilde{\beta}^T \tilde{g} - E[\tilde{\beta}^T \tilde{g}] \right)^2 \right] \\ &= E \left[\tilde{g} \left(\left(\tilde{\beta} - E[\tilde{\beta}] \right)^2 \right)^T \tilde{g} \right] \\ &= \tilde{g}^T \text{cov}(\tilde{\beta}) \tilde{g} \\ &= \tilde{g}^T (\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T)^{-1} \tilde{g} \sigma^2. \end{aligned} \quad (6)$$

The proof is finished by showing that $\tilde{g}^T (\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T)^{-1} \tilde{g} \leq 1$ if \mathcal{J}_g is an enclosing neighborhood.

Assume \mathcal{J}_g is an enclosing neighborhood. Then by definition it must be that $|\mathcal{J}_g| \geq d + 1$, and that there exists some weight vector v such that $\tilde{\mathbf{X}}v = \tilde{g}$ (which includes the constraint that $\mathbf{1}^T v = 1$) and $v \succeq 0$. The training and test samples are assumed to be drawn *iid* from a sufficiently nice distribution over the d -dimensional feature space such that the training and test samples are in general position with probability one; that is, the enclosing neighbors and test sample do not lie in a degenerate subspace, and, thus, it must be that the matrix $\tilde{\mathbf{X}}$ is full rank. Then the Moore–Penrose pseudo-inverse of $\tilde{\mathbf{X}}$ is well defined as the $m \times 1$ vector $w = \tilde{\mathbf{X}}^T(\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T)^{-1}\tilde{g}$, and w is the minimum norm solution to $\tilde{\mathbf{X}}w = \tilde{g}$ such that $w^T w \leq v^T v$ for any v that satisfies $\tilde{\mathbf{X}}v = \tilde{g}$ [30].

Then

$$\begin{aligned} w^T w &= \tilde{g}^T (\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T)^{-1} \tilde{\mathbf{X}}\tilde{\mathbf{X}}^T (\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T)^{-1} \tilde{g} \\ &= \tilde{g}^T (\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T)^{-1} \tilde{g}. \end{aligned} \quad (7)$$

Because $0 \leq v_j \leq 1$ for each $j \in \mathcal{J}_g$, it must be that $0 \leq v_j^2 \leq v_j$. Combining these facts with the property that $w^T w \geq 0$ because it is a sum of squared elements, the following holds:

$$0 \leq w^T w \leq v^T v = \sum_{j=1}^m v_j^2 \leq \sum_{j=1}^m v_j = 1.$$

Then from (7) it must also be that $\tilde{g}^T (\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T)^{-1} \tilde{g} \leq 1$, which coupled with (6) completes the proof.

APPENDIX B

METHOD FOR CALCULATING THE ENCLOSING k-NN NEIGHBORHOOD

- 1) Define an ϵ that is the threshold for how small the distance to enclosure must be before considering the neighborhood to effectively enclose the test point in its convex hull. Generally, ϵ should be small, but how small may depend on the relative scale of the data. For the CIELAB space, where a *just noticeable difference* is roughly $2\Delta E$, we set $\epsilon = .1$.

- 2) Re-order the set of training samples $\{x_i\}$ for $i = 1, \dots, n$ by distance from the test point g so that x_j is the j th NN to g .
- 3) Add x_1 to the set \mathcal{S} .
- 4) Define the indicator function $I(x_j)$, where $I(x_j) = 1$ if x_j lies in the same half-space as g with respect to the hyperplane that passes through x_1 and is normal to the vector connecting g to x_1 , and $I(x_j) = 0$ otherwise.
- 5) Add to the set \mathcal{S} the training point x_j^* nearest to g in the half-space, that is

$$x_j^* = \arg \min_{x_i: I(x_i)=1} \|g - x_i\|_2.$$

- 6) If the distance to enclosure $D(g, \mathcal{S}) < \epsilon$, then stop iterating, and the set of all training samples closer than x_j^* to g form the enclosing k-NN neighborhood.
- 7) Project g onto the convex hull of \mathcal{S} , and denote this point \hat{g} . Re-define the indicator function $I(x_j) = 1$ if x_j lies in the same half-space as g with respect to the hyperplane that passes through \hat{g} and is normal to the vector connecting g to \hat{g} , and $I(x_j) = 0$ otherwise. If $I(x_j) = 0$ for all training samples, then stop iterating, and the set of all training samples that are closer than the farthest member of \mathcal{S} form the enclosing k-NN neighborhood.
- 8) Repeat steps 5)–7) until a stopping criteria is met.

APPENDIX C

PROOF OF THEOREM 2

To prove the theorem, the following lemma will be used.

Lemma: Given $\mathbf{X} \in \mathbb{R}^{n \times d}$, let $\text{conv}(\mathbf{X})$ denote the convex hull of the rows $\{x_i\}_{i=1:n}$ of \mathbf{X} . If and only if the origin $\mathbf{0} \in \text{conv}(\mathbf{X})$, then the origin is in the convex hull of some positive scaling of the $\{x_i\}_{i=1:n}$, i.e., $\mathbf{0} \in \text{conv}(\mathbf{A}\mathbf{X})$, where \mathbf{A} is a positive definite diagonal $n \times n$ matrix.

Proof of Lemma: Suppose $\mathbf{0} \in \text{conv}(\mathbf{X})$. By definition, there exists a weight vector w such that $w^T \mathbf{1} = 1$, $w \succeq 0$, and $w^T \mathbf{X} = \mathbf{0}$. If \mathbf{X} is scaled by the positive definite diagonal matrix \mathbf{A} , then it must be shown that there exists a set of weights w' with the properties that $w'^T \mathbf{1} = 1$, $w' \succeq 0$, and $w'^T \mathbf{A}\mathbf{X} = \mathbf{0}$. Denote the normalization scalar $z = w^T \mathbf{A}^{-1} \mathbf{1}$, then it can be seen that one such weight vector that satisfies these conditions is $w' = (\mathbf{A}^{-1} w) / z$ and we conclude that $\mathbf{0} \in \text{conv}(\mathbf{A}\mathbf{X})$. Next, suppose that $\mathbf{0} \notin \text{conv}(\mathbf{X})$, then it must be shown that scaling \mathbf{X} by any positive definite diagonal matrix \mathbf{A} does not form a convex hull that contains the origin. The proof is by contradiction: assume that $\mathbf{0} \in \text{conv}(\mathbf{A}\mathbf{X})$ but $\mathbf{0} \notin \text{conv}(\mathbf{X})$. The first part of this proof could be applied, scaling $\mathbf{A}\mathbf{X}$ by \mathbf{A}^{-1} , which would lead to the conclusion that $\mathbf{0} \in \text{conv}(\mathbf{X})$, thus forming a contradiction.

Now we begin the body of the proof of Theorem 2. Without loss of generality, assume the test point is the origin $g = \mathbf{0}$. Let \mathbf{X} be the random $n \times d$ matrix with rows $\{x_i\}_{i=1:n}$ drawn independently and identically from a symmetric distribution in \mathbb{R}^d . Rearrange the rows of \mathbf{X} so that they are sorted such that $\|x_{k-1}\|_2 \leq \|x_k\|_2 \leq \|x_{k+1}\|_2$ for all k . As established in the lemma, without a loss of generality with respect to the event $\mathbf{0} \in \text{conv}(\mathbf{X})$, scale all rows such that $\|x_j\|_2 = 1$ for all j .

Then $\mathbf{0} \in \text{conv}(\mathbf{X})$ if and only if $n > d$ and the row vectors are not all contained in some hemisphere [31]. Let H_n indicate the event that n vectors lie on the same hemisphere, and \bar{H}_n will denote the complement of H_n . Wendel [32] showed that for n points chosen uniformly on a hypersphere in \mathbb{R}^d

$$P(H_n) = 2^{-n+1} \sum_{k=0}^{d-1} \binom{n-1}{k} \quad \forall n \geq 1. \quad (8)$$

Let F_n be the event that: the first n ordered points enclose the origin, but the first $n-1$ ordered points do not enclose the origin. The probability of the event F_n is

$$\begin{aligned} P(F_n) &= P(\bar{H}_n, H_{n-1}) \\ &= P(\bar{H}_n | H_{n-1}) P(H_{n-1}) \\ &= (1 - P(H_n | H_{n-1})) P(H_{n-1}) \\ &= P(H_{n-1}) - P(H_n, H_{n-1}) \\ &= P(H_{n-1}) - P(H_n). \end{aligned} \quad (9)$$

Because one or zero points cannot complete a convex hull around the origin, $P(F_0) = 0$ and $P(F_1) = 0$. Combining (8) and (9), and using the recurrence relation of the binomial coefficient

$$\binom{n}{k} + \binom{n}{k+1} = \binom{n+1}{k+1} \quad (10)$$

$$\begin{aligned} P(F_n) &= 2^{-n+2} \sum_{k=0}^{d-1} \binom{n-2}{k} - 2^{-n+1} \sum_{k=0}^{d-1} \binom{n-1}{k} \\ &= 2^{-n+1} \sum_{k=0}^{d-1} \left[2 \binom{n-2}{k} - \binom{n-1}{k} \right] \\ &= 2^{-n+1} \sum_{k=0}^{d-1} \left[2 \binom{n-2}{k} - \binom{n-2}{k-1} - \binom{n-2}{k} \right] \\ &= 2^{-n+1} \left[\binom{n-2}{d-1} - \binom{n-2}{d-2} + \binom{n-2}{d-2} \right. \\ &\quad \left. - \binom{n-2}{d-3} + \dots + \binom{n-2}{0} - \binom{n-2}{-1} \right] \\ &= 2^{-n+1} \left[\binom{n-2}{d-1} - \binom{n-2}{-1} \right] \\ &= 2^{-n+1} \binom{n-2}{d-1} \quad \text{for all } n \geq 2 \end{aligned}$$

where the last line follows because $\binom{r}{-1} = 0$ for all r [33, p. 154]. Then

$$E[F_a] = \sum_{a=2}^{\infty} a P(F_a) = \sum_{a=2}^{\infty} a 2^{-a+1} \binom{a-2}{d-1}.$$

To simplify, change variables to $b = a - 2$

$$\begin{aligned} E[F_a] &= \sum_{b=0}^{\infty} b 2^{-b-1} \binom{b}{d-1} + 2^{-b} \binom{b}{d-1} \\ &= \sum_{b=0}^{\infty} 2^{-b-1} \left[b \binom{b}{d-1} + 2 \binom{b}{d-1} \right] \\ &= \frac{bb!}{(d-1)!(b-d+1)!} + \frac{2b!}{(d-1)!(b-d+1)!} \\ &= \frac{(b+1)b!}{(d-1)!(b-d+1)!} + \frac{b!}{(d-1)!(b-d+1)!} \\ &= \frac{d(b+1)!}{d!(b-d+1)!} + \frac{b!}{(d-1)!(b-d+1)!} \\ &= \left[d \binom{b+1}{d} + \binom{b}{d-1} \right] \\ &= d \binom{b+1}{d} + \binom{b+1}{d} - \binom{b}{d} \\ &= \sum_{b=0}^{\infty} 2^{-b-1} \left[(d+1) \binom{b+1}{d} - \binom{b}{d} \right] \end{aligned}$$

where the second to last line follows from (10). Expand the recurrence

$$\begin{aligned} E[F_a] &= \lim_{n \rightarrow \infty} \left(2^{-1}(d+1) \binom{1}{d} - 2^{-1} \binom{0}{d} \right. \\ &\quad \left. + 2^{-2}(d+1) \binom{2}{d} - 2^{-2} \binom{1}{d} + \dots \right. \\ &\quad \left. + 2^{-n}(d+1) \binom{n}{d} - 2^{-n} \binom{n-1}{d} \right. \\ &\quad \left. + 2^{-n-1}(d+1) \binom{n+1}{d} - 2^{-n-1} \binom{n}{d} \right) \\ &= \lim_{n \rightarrow \infty} \left(-2^{-1} \binom{0}{d} + 2^{-2}(2(d+1)-1) \binom{1}{d} + \dots \right. \\ &\quad \left. + 2^{-n-1}(2(d+1)-1) \binom{n}{d} \right. \\ &\quad \left. + 2^{-n-1}(d+1) \binom{n+1}{d} \right) \\ &= \lim_{n \rightarrow \infty} \left(-2^{-1} \binom{0}{d} + \left(\sum_{i=1}^n 2^{-i-1}(2d+1) \binom{i}{d} \right) \right. \\ &\quad \left. + 2^{-n-1}(d+1) \binom{n+1}{d} \right). \end{aligned}$$

The first and third terms in this equation converge to zero as $n \rightarrow \infty$, leaving

$$E[F_a] = (2d+1) \sum_{i=1}^{\infty} 2^{-i-1} \binom{i}{d}.$$

Using the identity $\binom{0}{d} = 0$ [33, p. 155], and the summation [33, p. 199]

$$\frac{z^d}{(1-z)^{d+1}} = \sum_{i=0}^{\infty} \binom{i}{d} z^i$$

with $z = 1/2$, establishes the result: $E[F_a] = 2d + 1$.

ACKNOWLEDGMENT

The authors would like to thank R. Bala, S. Upton, and J. Bowen for helpful discussions.

REFERENCES

- [1] W. Lam, C. Keung, and D. Liu, "Discovering useful concept prototypes for classification based on filtering and abstraction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 8, pp. 1075–1090, Aug. 2002.
- [2] R. C. Holte, "Very simple classification rules perform well on most commonly used data sets," *Mach. Learn.*, vol. 11, pp. 63–90, 1993.
- [3] M. R. Gupta, R. Gray, and R. Olshen, "Nonparametric supervised learning by linear interpolation with maximum entropy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 5, pp. 766–781, May 2006.
- [4] C. Loader, *Local Regression and Likelihood*. New York: Springer, 1999.
- [5] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York: Springer-Verlag, 2001.
- [6] R. Bala, "Device characterization," in *Digital Color Handbook*, G. Sharma, Ed. Boca Raton, FL: CRC, 2003, ch. 5, pp. 269–384.
- [7] P. Emmel, "Physical models for color prediction," in *Digital Color Handbook*, G. Sharma, Ed. Boca Raton, FL: CRC, 2003, ch. 3, pp. 173–238.
- [8] B. Fraser, C. Murphy, and F. Bunting, *Real World Color Management*. Berkeley, CA: Peachpit, 2003.
- [9] M. R. Gupta and R. Bala, *Personal Communication With Raja Bala*. Jun. 21, 2006.
- [10] A. E. Hoerl and R. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, pp. 55–67, 1970.
- [11] K. Fukunaga and L. Hostetler, "Optimization of k -nearest neighbor density estimates," *IEEE Trans. Inf. Theory*, vol. IT-19, no. 3, pp. 320–326, Mar. 1973.
- [12] R. Short and K. Fukunaga, "The optimal distance measure for nearest neighbor classification," *IEEE Trans. Inf. Theory*, vol. IT-27, no. 5, pp. 622–627, May 1981.
- [13] K. Fukunaga and T. Flick, "An optimal global nearest neighbor metric," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, no. 6, pp. 314–318, Jun. 1984.
- [14] J. Myles and D. Hand, "The multi-class metric problem in nearest neighbour discrimination rules," *Pattern Recognit.*, vol. 23, pp. 1291–1297, 1990.
- [15] J. Friedman, "Flexible metric nearest neighbor classification," Tech. Rep., Stanford Univ., Stanford, CA, 1994.
- [16] C. Domeniconi, J. Peng, and D. Gunopulos, "Locally adaptive metric nearest neighbor classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1281–1285, Sep. 2002.
- [17] R. Paredes and E. Vidal, "Learning weighted metrics to minimize nearest-neighbor classification error," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 7, pp. 1100–1110, Jul. 2006.
- [18] T. Hastie and R. Tibshirani, "Discriminative adaptive nearest neighbour classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 6, pp. 607–615, Jun. 1996.
- [19] D. Wettschereck, D. W. Aha, and T. Mohri, "A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms," *Artif. Intell. Rev.*, vol. 11, pp. 273–314, 1997.
- [20] J. Peng, D. R. Heisterkamp, and H. K. Dai, "Adaptive quasiconformal kernel nearest neighbor classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 5, pp. 656–661, May 2004.
- [21] R. Nock, M. Sebban, and D. Bernard, "A simple locally adaptive nearest neighbor rule with application to pollution forecasting," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 17, no. 8, pp. 1–14, 2003.
- [22] J. Zhang, Y.-S. Yim, and J. Yang, "Intelligent selection of instances for prediction functions in lazy learning algorithms," *Artif. Intell. Rev.*, vol. 11, pp. 175–191, 1997.
- [23] B. Bhattacharya, K. Mukherjee, and G. Toussaint, "Geometric decision rules for instance-based learning," *Lecture Notes Comput. Sci.*, pp. 60–69, 2005.
- [24] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*. New York: Springer-Verlag, 1996.
- [25] P. Hall, J. S. Marron, and A. Neeman, "Geometric representation of high dimension, low sample size data," *J. Roy. Statist. Soc. B*, vol. 67, pp. 427–444, 2005.
- [26] R. Sibson, "A brief description of natural neighbour interpolation," in *Interpreting Multivariate Data*, V. Barnett, Ed. New York: Wiley, 1981, pp. 21–36.
- [27] A. Okabe, B. Boots, K. Sugihara, and S. Chiu, *Spatial Tessellations*. Chichester, U.K.: Wiley, 2000, ch. 6, pp. 418–421.
- [28] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Trans. Math. Softw.*, vol. 22, no. 4, pp. 469–483, 1996.
- [29] H. Kang, *Color Technology for Electronic Imaging Devices*. Bellingham, WA: SPIE, 1997.
- [30] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2006.
- [31] R. Howard and P. Sisson, "Capturing the origin with random points: Generalizations of a Putnam problem," *College Math. J.*, vol. 27, no. 3, pp. 186–192, May 1996.
- [32] J. Wendel, "A problem in geometric probability," *Math. Scand.*, vol. 11, pp. 109–111, 1962.
- [33] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics*. New York: Addison-Wesley, 1989.



Maya R. Gupta (M'04) received the B.S. degree in electrical engineering and the B.A. degree in economics from Rice University, Houston, TX, in 1997, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 2003, as a National Science Foundation Graduate Fellow.

From 1999–2003, she was with Ricoh's California Research Center as a color image processing Research Engineer. In the fall of 2003, she joined the faculty of the Department of Electrical Engineering, University of Washington, Seattle, as an Assistant Professor, and also became an Adjunct Assistant Professor of applied mathematics in 2005.

Prof. Gupta was awarded the 2007 Office of Naval Research Young Investigator Award and the 2007 University of Washington Department of Electrical Engineering Outstanding Teaching Award.



Eric K. Garcia received the B.S. degree in computer engineering from Oregon State University, Corvallis, in 2004, as a recipient of the Gates Millennium Scholarship, and the M.S. degree in electrical engineering from the University of Washington, Seattle, in 2006, as a GEM Master's Fellow. He is currently pursuing the Ph.D. degree in electrical engineering at the University of Washington as a GEM Ph.D. Fellow.



Erika Chin received the B.S. degree in computer science from the University of Virginia, Charlottesville, in 2007, and is currently pursuing the Ph.D. degree in computer science at the University of California, Berkeley.