A Statistical, Nonparametric Methodology for Document Degradation Model Validation

Tapas Kanungo, *Member*, *IEEE*, Robert M. Haralick, *Fellow*, *IEEE*,
Henry S. Baird, *Senior Member*, *IEEE*, Werner Stuezle, and David Madigan

Abstract—Printing, photocopying, and scanning processes degrade the image quality of a document. Statistical models of these degradation processes are crucial for document image understanding research. Models allow us to predict system performance, conduct controlled experiments to study the breakdown points of the systems, create large multilingual data sets with groundtruth for training classifiers, design optimal noise removal algorithms, choose values for the free parameters of the algorithms, and so on. Although research in document understanding started many decades ago, only two document degradation models have been proposed thus far. Furthermore, no attempts have been made to statistically validate these models. In this paper, we present a statistical methodology that can be used to validate local degradation models. This method is based on a nonparametric, two-sample permutation test. Another standard statistical device—the power function—is then used to choose between algorithm variables such as distance functions. Since the validation and the power function procedures are independent of the model, they can be used to validate any other degradation model. A method for comparing any two models is also described. It uses p-values associated with the estimated models to select the model that is closer to the real world.

Index Terms—Model validation, nonparametric statistical tests, permutation tests, document degradation models, simulation models, OCR.

1 Introduction

Printing, photocopying, and scanning processes degrade the image quality of any document. Statistically valid models of these degradation processes can impact document image understanding research in many ways. Degradation models can be used to conduct controlled experiments to study the breakdown points of OCR systems, create large multilingual data sets with groundtruth for training classifiers, design optimal noise removal algorithms, choose values for the free parameters of the algorithms, predict OCR performance, and so on. Whereas research in document understanding started decades ago, only two document degradation models have been proposed thus far. Furthermore, no attempts have been made to statistically validate these models.

The current OCR evaluation methods rely on scanned documents, corresponding hand-entered ASCII groundtruth strings, and string matching algorithms that compare the groundtruth string against the OCR-generated string. The errors in the groundtruth are reduced by a process of cross-checking. This method is very expensive, laborious, and prone to errors. Furthermore, since the datasets are expensive, it is not possible to create large datasets that are representative of the variety of layout, font, and degradation levels seen in real-world documents. Despite these problems, various document databases with groundtruth have been created.

Our methodology for characterizing OCR algorithms is based on evaluating the algorithms on synthetically degraded documents. First, a word processor is used to create an ideal document in any language, format, or style. A bitmap version of this document is then created and degraded using a computer model of the real degradation process. This method has many advantages. First, since the ideal document is created using a word processor, the groundtruth information associated with each character—-location, identity, font type, etc.—is known without error. Second, the word processor can be used to reformat the documents (two columns, one column, various font types, sizes, etc.) to study the sensitivity of the OCR algorithm to these variables. Third, since the degradation model is under our control, we can create documents with varying levels of degradation and study how and where the OCR algorithm breaks down. Fourth, sample size is not a problem at all—any number of degraded samples can be created since all that needs to be done is to simulate another set of characters. In addition, there is no dearth of formatted documents—we create such documents daily, and so do academic journal publishers. Fifth, the model itself can be used in creating noise removal algorithms, training classifiers, choosing algorithm parameters, etc.

The main drawback of the above methodology is that it relies heavily on the simulation model being correct. That is,

- *T. Kanungo is with the Language and Media Processing Lab, Center for Automation Research, University of Maryland, 3453 A.V. Williams, College Park, MD 20742. E-mail: kanungo@cfar.umd.edu.*
- *R. Haralick and W. Stuezle are with the Deptartment of Electrical Engineering, University of Washington, Seattle, WA 98195. E-mail: {haralick@ee, wxs@stat}.washington.edu.*
- *H. Baird is with Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304. E-mail: baird@parc.xerox.com.*
- *D. Madigan is with Soliloquy Inc., 255 Park Ave., 6th Floor, New York, NY 10010. E-mail: dmadigan@soliloquy.com.*

Manuscript received 15 Jan. 1999; revised 12 June 2000; accepted 21 July 2000.
Recommended for acceptance by D. Dori.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 108985.

it assumes that the simulation model closely mimics reality. Thus, it is imperative that we validate the degradation model against real data. Only then can the simulations be used *in place of* real data. If the degradation model is not validated, results on the synthetically degraded documents should be used with caution, though they are still useful since they give *some* indication about the performance of the OCR algorithm.

In this paper, we present a statistical methodology that can be used to validate the local degradation models. This method is based on a nonparametric, two-sample permutation test. Another standard statistical device—the power function—is then used to choose between algorithm variables such as distance functions. Since the validation and the power function procedures are independent of the model, they can be used to validate any other degradation model. A method of comparing any two models is also described. It uses p-values associated with the estimated models to select the model that is closer to the real world.

In Section 2, we survey the related literature in the areas of degradation models, model validation, and statistical hypothesis testing and discuss their shortcomings. In Section 3, we describe our document degradation model for the local distortions that occur while printing, photocopying, and scanning documents. The model is independent of the language in which the document is written. Our validation methodology is described in Section 4 and in Section 5, we apply it to datasets with known distributions to understand the performance of the permutation tests. In Section 6, we give results of validation experiments on document images and, in Section 7, we present conclusions.

2 Related Literature

Most OCR algorithms make use of explicit or implicit models of degradation. The authors, however, have not proposed statistical methods for validating or comparing these models. An explicit degradation model that was recently proposed is that of Baird [2], [3], [4]. Unfortunately, his degradation model is not validated either. Furthermore, his paper advocates the use of isolated, synthetically degraded characters. Thus, the degradation due to merging of neighboring characters is not reflected in his model. In addition, the unigram and bigram occurrence probabilities of characters in real-world text are not reflected in isolated-character experiments.

In contrast, our document degradation model, which is described in Section 3, advocates the use of complete documents for generating synthetically degraded characters. Thus, it takes into account the degradations arising due to merging of characters, the occurrence probabilities of individual characters, and the variability in the layout structure of the documents. The pixel degradations themselves are based on a local morphological model, which models the final spatial characteristics of the degradation process rather than the underlying physical process.

To the best of our knowledge, the only other work on validation of document degradation models is that of Nagy [17], and Li et al. [15], [16]. They are of the opinion that a degradation model is valid if the OCR confusion matrices that result from synthetically degraded documents are similar to the OCR confusion matrices produced from real documents. Unfortunately, this methodology validates the model-OCR combination and not the model itself. For instance, if the OCR system automatically filters noise, their validation process will not detect any difference between the real documents and the synthetically degraded documents even if the degradation process adds noise to the document. Furthermore, although they treat the OCR as a black box, the OCR algorithm itself has many parameters that can greatly influence the decisions of the validation procedure. Another drawback of their approach is that they do not indicate how their validation procedure can be compared to other validation procedures.

Our validation method, on the other hand, reduces the problem of model validation to a nonparametric statistical hypothesis testing problem, which is a well-studied and accepted method in statistics [7], [6]. In addition, we use simple character distance functions for the validation procedure, instead of entire OCR systems. Although the validation process now depends on these distance functions, they are much simpler than OCR black boxes. Finally, we provide a technique for comparing our validation method with other validation methods. This comparison procedure is based on "power functions," which again are standard statistical devices for comparing hypothesis testing procedures.

3 A Document Degradation Model

In this section, we describe a document degradation model for local distortions that are introduced during the printing, photocopying, and scanning processes. A model for the perspective and illumination distortions that get introduced when we photocopy or scan thick bound books is described in [11], [12], [8].

Our local document degradation model accounts for 1) pixel inversion (from foreground to background and vice versa) that occurs independently at each pixel due to light intensity fluctuations, sensitivity of the sensors, and the thresholding level and 2) blurring that occurs due to the point-spread function of the scanner optical system. We model the pixel-flipping probability of a background pixel as an exponential function of its distance from the nearest boundary pixel. The parameter α_0 is the initial value for the exponential and the decay speed of the exponential is controlled by the parameter α. The foreground and background four-neighbor distance are computed using a standard distance transform algorithm (see [5]). The flipping probabilities of the foreground pixels are similarly controlled by β_0 and β. The parameter η is the constant probability of flipping for all pixels. Finally, the last parameter k, which is the size of the disk used in the morphological closing operation, accounts for the correlation introduced by the point-spread function of the optical system.

Thus, the degradation model has six parameters: $\Theta = (\eta, \alpha_0, \alpha, \beta_0, \beta, k)$. These parameters are used to degrade an ideal binary image as follows:

1. Compute the distance d of each pixel from the character boundary.

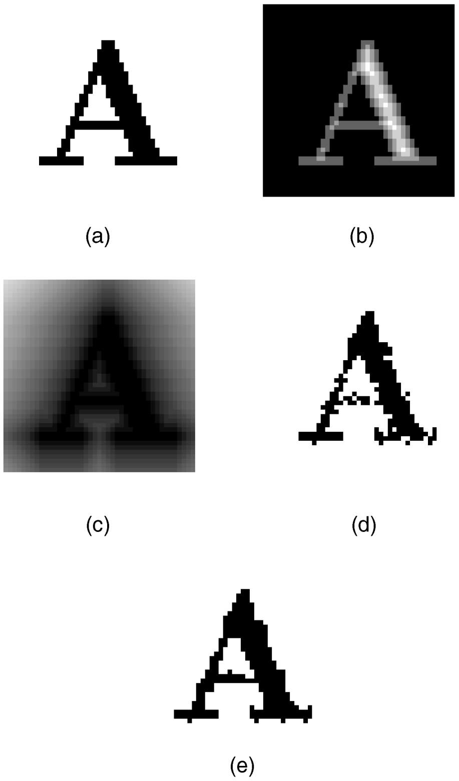


Fig. 1. Local document degradation model: (a) Ideal noise-free character. (b) Distance transform of the foreground. (c) Distance transform of the background. (d) Result of the random pixel-flipping process (the probability of a pixel flipping is $p(0|d,\beta,f) = p(1|d,\alpha,b) = \alpha_0 e^{-\alpha d^2}$; here, $\alpha = \beta = 2, \alpha_0 = \beta_0 = 1$). (e) Morphological closing of the result in (d) by a 2×2 binary structuring element.

2. Flip each foreground pixel with probability $p(0|1,d,\alpha_0,\alpha) = \alpha_0 e^{-\alpha d^2} + \eta$.
3. Flip each background pixel with probability $p(1|0,d,\beta_0,\beta) = \beta_0 e^{-\beta d^2} + \eta$.
4. Perform a morphological closing operation with a disk structuring element of diameter k.

Software for simulating noisy documents using the above degradation model is available from the University of Washington English Document Database I and the model has also been described in the literature [12], [11]. The application of the various steps of the model is illustrated in Fig. 1. In Fig. 1a, we show an ideal character. The distance transform of the foreground of Fig. 1a is shown in Fig. 1b. The brighter pixels are further away from the pixel boundary. The distance transform of the background is shown in Fig. 1d. The ideal image after its pixels have been flipped according to the model is shown in Fig. 1d. The final image after the closing operation is shown in Fig. 1e.

The procedure described above works on bit-mapped images. Since there is no restriction on the size of the image that can be degraded, or the language of the written text, an entire document page image can be degraded using this model. In fact, since typesetting is under the experimenter's control, the same text can be reformatted in various styles (single column, multiple column, report, book, etc.), font types (Roman, Helvetica, etc.), and font sizes (9pt, 10pt, 12pt, etc.). Thus, the performance of any character recognition system can be studied by providing as input the same (or different) text formatted in various styles with varied but controlled degradation.

Now, we show examples where we degrade complete document pages using our degradation model. In Fig. 2a, we show an ideal document formatted in LaTex, using the IEEE Transactions typesetting style. In Fig. 2b, we show a degraded version of the document in Fig. 2a.

The noise-free documents are typeset using the LaTeX formatting system [14], [13]. The ASCII files containing the text and the LaTeX typesetting information are then converted into a device-independent format (DVI) using LaTeX. A program called DVI2TIFF—which is a modified version of a DVI file previewer called XDVI [19]—is run to produce one bit/pixel binary images in TIFF format from the DVI files. Besides producing the binary images of the documents, DVI2TIFF also produces the groundtruth information regarding each character in the document image.

The local document degradation model is another program called DDM. This program takes as input an ideal binary document image in TIFF format and a file containing the degradation model parameter values and produces the binary degraded images in TIFF format.

Both programs—DVI2TIFF and DDM—are implemented using the C language and have been tested on SUN and IBM machines running the UNIX operating system. The software is available on the UW CD-ROM-1 [18].

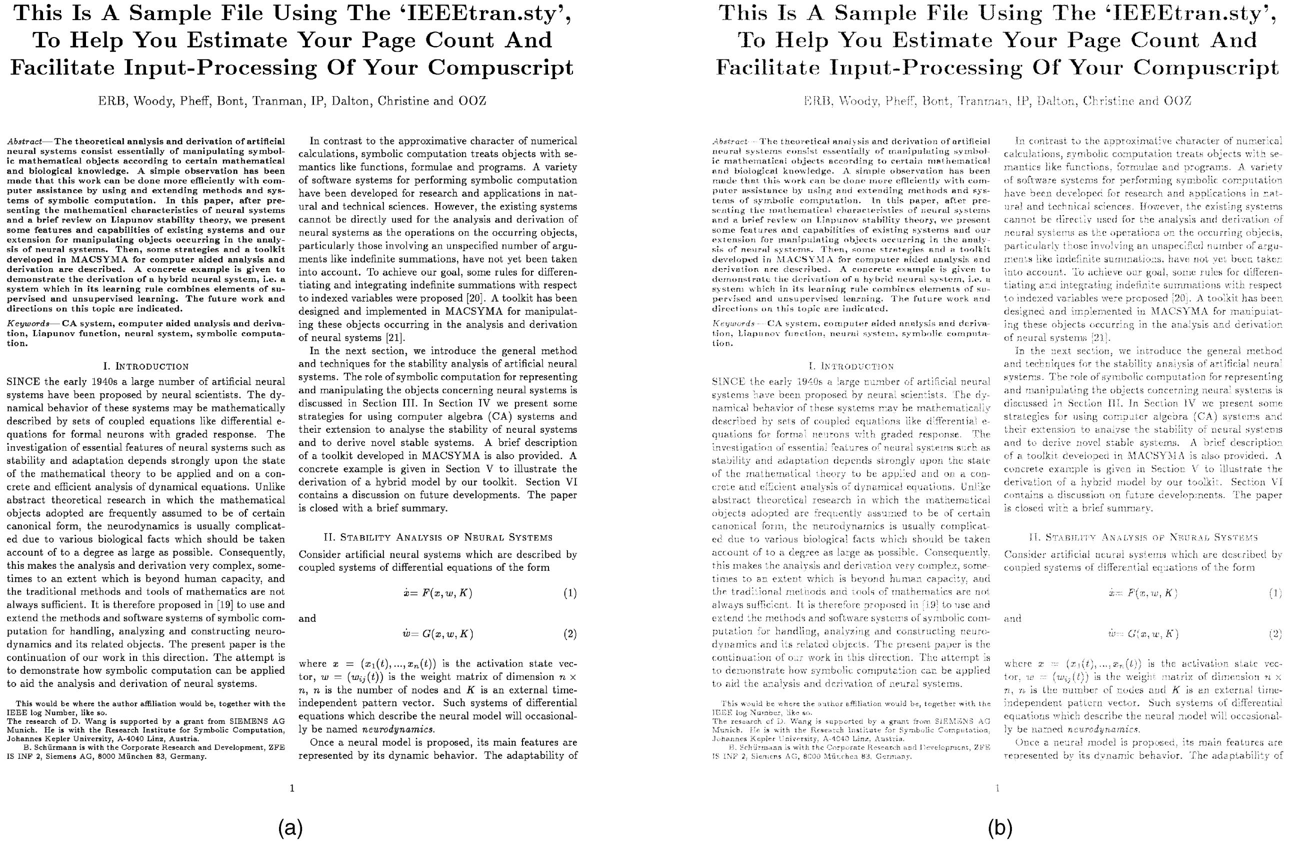
4 Model Validation and Parameter Estimation

4.1 Statistical Problem Definition

In this section, we formulate the degradation model validation problem as a statistical problem. Although degradation of the document is over the entire page, the degradation process itself is local. That is, degradation in one region does not influence the degradation process in another sufficiently distant region. More precisely, the degradation at a pixel is influenced only by pixels within a local neighborhood. Thus, one way to characterize the degradation process is to study the degradation of local patterns. Since the most common patterns that occur on a document page are characters, we statistically characterize the degradation of individual characters on the page and use this characterization to estimate the parameters of a degradation model that produces similar degradations.

Assume that a scanned character is represented by a 30×30 matrix of zeros and ones. This matrix can be represented as $1,000 \times 1$ vector x $(30 \times 30 \approx 1,000)$. Let B be the space of D = 1,000-dimensional binary vectors, that is, $B = \{0,1\}^D$. Now, let $x_1, x_2, \ldots, x_N \in B$ be independent and identically distributed D-dimensional vectors representing instances of degraded characters produced from the same class ω. That is, each x_i is a degraded character that is produced from the same ideal pattern ω (say the ideal character "e") and the same degradation process. The validation problem we need to address is:

> Suppose we are given a set of *real* degraded instances $x_1, \ldots, x_N \in B$ of the pattern ω and another set of *synthetic*



This Is A Sample File Using The 'IEEEtran.sty', To Help You Estimate Your Page Count And Facilitate Input-Processing Of Your Compuscript

ERB, Woody, Pheff, Bont, Tranman, IP, Dalton, Christine and OOZ

Abstract—**The theoretical analysis and derivation of artificial neural systems consist essentially of manipulating symbolic mathematical objects according to certain mathematical and biological knowledge. A simple observation has been made that this work can be done more efficiently with computer assistance by using and extending methods and systems of symbolic computation. In this paper, after presenting the mathematical characteristics of neural systems and a brief review on Liapunov stability theory, we present some features and capabilities of existing systems and our extension for manipulating objects occurring in the analysis of neural systems. Then, some strategies and a toolkit developed in MACSYMA for computer aided analysis and derivation are described. A concrete example is given to demonstrate the derivation of a hybrid neural system, i.e. a system which in its learning rule combines elements of supervised and unsupervised learning. The future work and directions on this topic are indicated.**

Keywords—**CA system, computer aided analysis and derivation, Liapunov function, neural system, symbolic computation.**

I. Introduction

SINCE the early 1940s a large number of artificial neural systems have been proposed by neural scientists. The dynamical behavior of these systems may be mathematically described by sets of coupled equations like differential equations for formal neurons with graded response. The investigation of essential features of neural systems such as stability and adaptation depends strongly upon the state of the mathematical theory to be applied and on a concrete and efficient analysis of dynamical equations. Unlike abstract theoretical research in which the mathematical objects adopted are frequently assumed to be of certain canonical form, the neurodynamics is usually complicated due to various biological facts which should be taken account of to a degree as large as possible. Consequently, this makes the analysis and derivation very complex, sometimes to an extent which is beyond human capacity, and the traditional methods and tools of mathematics are not always sufficient. It is therefore proposed in [19] to use and extend the methods and software systems of symbolic computation for handling, analyzing and constructing neurodynamics and its related objects. The present paper is the continuation of our work in this direction. The attempt is to demonstrate how symbolic computation can be applied to aid the analysis and derivation of neural systems.

In contrast to the approximative character of numerical calculations, symbolic computation treats objects with semantics like functions, formulae and programs. A variety of software systems for performing symbolic computation have been developed for research and applications in natural and technical sciences. However, the existing systems cannot be directly used for the analysis and derivation of neural systems as the operations on the occurring objects, particularly those involving an unspecified number of arguments like indefinite summations, have not yet been taken into account. To achieve our goal, some rules for differentiating and integrating indefinite summations with respect to indexed variables were proposed [20]. A toolkit has been designed and implemented in MACSYMA for manipulating these objects occurring in the analysis and derivation of neural systems [21].

In the next section, we introduce the general method and techniques for the stability analysis of artificial neural systems. The role of symbolic computation for representing and manipulating the objects concerning neural systems is discussed in Section III. In Section IV we present some strategies for using computer algebra (CA) systems and their extension to analyse the stability of neural systems and to derive novel stable systems. A brief description of a toolkit developed in MACSYMA is also provided. A concrete example is given in Section V to illustrate the derivation of a hybrid model by our toolkit. Section VI contains a discussion on future developments. The paper is closed with a brief summary.

II. Stability Analysis of Neural Systems

Consider artificial neural systems which are described by coupled systems of differential equations of the form

$$\dot{x}= F(x, w, K) \quad (1)$$

and

$$\dot{w}= G(x, w, K) \quad (2)$$

where $x = (x_1(t), ..., x_n(t))$ is the activation state vector, $w = (w_{ij}(t))$ is the weight matrix of dimension $n \times n$, n is the number of nodes and K is an external time-independent pattern vector. Such systems of differential equations which describe the neural model will occasionally be named *neurodynamics*.

Once a neural model is proposed, its main features are represented by its dynamic behavior. The adaptability of

This would be where the author affiliation would be, together with the IEEE log Number, like so.
The research of D. Wang is supported by a grant from SIEMENS AG Munich. He is with the Research Institute for Symbolic Computation, Johannes Kepler University, A-4040 Linz, Austria.
B. Schürmann is with the Corporate Research and Development, ZFE IS INF 2, Siemens AG, 8000 München 83, Germany.

1

(a)

This Is A Sample File Using The 'IEEEtran.sty', To Help You Estimate Your Page Count And Facilitate Input-Processing Of Your Compuscript

ERB, Woody, Pheff, Bont, Tranman, IP, Dalton, Christine and OOZ

Abstract—**The theoretical analysis and derivation of artificial neural systems consist essentially of manipulating symbolic mathematical objects according to certain mathematical and biological knowledge. A simple observation has been made that this work can be done more efficiently with computer assistance by using and extending methods and systems of symbolic computation. In this paper, after presenting the mathematical characteristics of neural systems and a brief review on Liapunov stability theory, we present some features and capabilities of existing systems and our extension for manipulating objects occurring in the analysis of neural systems. Then, some strategies and a toolkit developed in MACSYMA for computer aided analysis and derivation are described. A concrete example is given to demonstrate the derivation of a hybrid neural system, i.e. a system which in its learning rule combines elements of supervised and unsupervised learning. The future work and directions on this topic are indicated.**

Keywords—**CA system, computer aided analysis and derivation, Liapunov function, neural system, symbolic computation.**

I. Introduction

SINCE the early 1940s a large number of artificial neural systems have been proposed by neural scientists. The dynamical behavior of these systems may be mathematically described by sets of coupled equations like differential equations for formal neurons with graded response. The investigation of essential features of neural systems such as stability and adaptation depends strongly upon the state of the mathematical theory to be applied and on a concrete and efficient analysis of dynamical equations. Unlike abstract theoretical research in which the mathematical objects adopted are frequently assumed to be of certain canonical form, the neurodynamics is usually complicated due to various biological facts which should be taken account of to a degree as large as possible. Consequently, this makes the analysis and derivation very complex, sometimes to an extent which is beyond human capacity, and the traditional methods and tools of mathematics are not always sufficient. It is therefore proposed in [19] to use and extend the methods and software systems of symbolic computation for handling, analyzing and constructing neurodynamics and its related objects. The present paper is the continuation of our work in this direction. The attempt is to demonstrate how symbolic computation can be applied to aid the analysis and derivation of neural systems.

In contrast to the approximative character of numerical calculations, symbolic computation treats objects with semantics like functions, formulae and programs. A variety of software systems for performing symbolic computation have been developed for research and applications in natural and technical sciences. However, the existing systems cannot be directly used for the analysis and derivation of neural systems as the operations on the occurring objects, particularly those involving an unspecified number of arguments like indefinite summations, have not yet been taken into account. To achieve our goal, some rules for differentiating and integrating indefinite summations with respect to indexed variables were proposed [20]. A toolkit has been designed and implemented in MACSYMA for manipulating these objects occurring in the analysis and derivation of neural systems [21].

In the next section, we introduce the general method and techniques for the stability analysis of artificial neural systems. The role of symbolic computation for representing and manipulating the objects concerning neural systems is discussed in Section III. In Section IV we present some strategies for using computer algebra (CA) systems and their extension to analyse the stability of neural systems and to derive novel stable systems. A brief description of a toolkit developed in MACSYMA is also provided. A concrete example is given in Section V to illustrate the derivation of a hybrid model by our toolkit. Section VI contains a discussion on future developments. The paper is closed with a brief summary.

II. Stability Analysis of Neural Systems

Consider artificial neural systems which are described by coupled systems of differential equations of the form

$$\dot{x}= F(x, w, K) \quad (1)$$

and

$$\dot{w}= G(x, w, K) \quad (2)$$

where $x = (x_1(t), ..., x_n(t))$ is the activation state vector, $w = (w_{ij}(t))$ is the weight matrix of dimension $n \times n$, n is the number of nodes and K is an external time-independent pattern vector. Such systems of differential equations which describe the neural model will occasionally be named *neurodynamics*.

Once a neural model is proposed, its main features are represented by its dynamic behavior. The adaptability of

This would be where the author affiliation would be, together with the IEEE log Number, like so.
The research of D. Wang is supported by a grant from SIEMENS AG Munich. He is with the Research Institute for Symbolic Computation, Johannes Kepler University, A-4040 Linz, Austria.
B. Schürmann is with the Corporate Research and Development, ZFE IS INF 2, Siemens AG, 8000 München 83, Germany.

1

(b)

Fig. 2. (a) An ideal document page typeset using LaTeX and IEEE Transactions typesetting style. (b) A synthetically degraded version of the document in (a).

degraded instances $y_1, \ldots, y_M \in B$ of the pattern ω. Test the null hypothesis that $y_1, \ldots, y_M$ and $x_1, \ldots, x_N$, are samples taken from the same underlying population, to a specified significance level ϵ.

In our case D is large, typically on the order of 1,000. Thus, the number of possible x_is is $2^{1,000}$, which is approximately equal to 10^{300}—a dauntingly large number. The available sample size, N, is typically on the order of 1,000. Thus, the x_is occupy the space B extremely sparsely, which implies that the density function cannot be estimated reliably from the sample. This fact prohibits us from performing any standard statistical test based on density estimates. In the next section, we describe a nonparametric method that overcomes this problem.

4.2 Permutation Tests and Model Validation

In this section, we describe a nonparametric validation procedure that can be used to statistically validate any document degradation model. Suppose we are given a set of real degraded characters $X = \{x_1, x_2, \ldots, x_N\}$, and another set of artificially degraded characters $Y = \{y_1, y_2, \ldots, y_M\}$ that were created by perturbing an ideal character with a document degradation model. We can assume that the characters x_i and y_i are binary matrices of size (approximately) 30×30. Note that every x_i and y_i can be of different size because the scanned characters can be of different sizes. The question that needs to be addressed is whether or not the x_is and the y_is come from the same underlying population. At this point, it does not matter where the x_is and the y_is came from, they could both be synthetically generated, or both be real instances, or one of them could be synthetic and the other real. A statistical hypothesis test can be performed to test the null hypothesis that the underlying population distributions of the x_is and y_is are the same.

Standard parametric hypothesis testing procedures are not usable for our problem because the sizes of binary matrices x_i and y_i are not fixed. Furthermore, the size of the space to which they belong is very large (approximately 2^{900} if we assume each character to be of size 30×30) and so while in principle it is possible to estimate the density function, in practice it is not possible to do so because of the small sample size. Instead, we now describe a *nonparametric permutation test* (see [7], [6]) that performs this hypothesis test.

1. Given
   a. the real data $X = \{x_1, x_2, \ldots, x_N\}$,
   b. the synthetic data $Y = \{y_1, y_2, \ldots, y_M\}$,
   c. a distance function $\rho(X, Y)$ on sets,
   d. a distance function $\delta(x, y)$ on characters, and
   e. the size ϵ of the test (i.e., misdetection rate = ϵ).
2. Compute $d_0 = \rho(X, Y)$.
3. Create a new sample $Z = \{x_1, \ldots, x_N, y_1, \ \ldots, y_M\}$. Thus, Z has $N + M$ elements labeled z_i, $i = 1, \ldots, N + M$.
4. Randomly permute (reorder) Z.

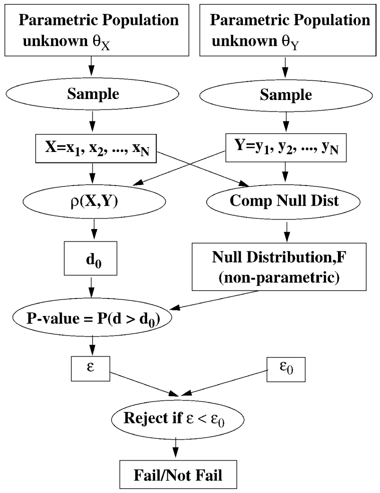

Parametric Population unknown θ_X
Parametric Population unknown θ_Y
Sample
Sample
$X=x_1, x_2, ..., x_N$
$Y=y_1, y_2, ..., y_N$
$\rho(X,Y)$
Comp Null Dist
d_0
Null Distribution,F (non-parametric)
P-value = $P(d > d_0)$
ε
ε_0
Reject if $\varepsilon < \varepsilon_0$
Fail/Not Fail

Fig. 3. Here, we show how the nonparametric test works when the two samples X and Y are from arbitrary distributions. For our problem, x_i and y_i are binary characters. In this case, the null distribution cannot be determined theoretically.

5. Partition the set Z into two sets X' and Y', where $X' = \{z_1, \ldots, z_N\}$ and $Y' = \{z_{N+1}, \ldots, z_{N+M}\}$.
6. Compute $d_i = \rho(X', Y')$.
7. Repeat steps 4, 5, and 6 K times to get K distances $d_1, \ldots, d_K$.
8. Compute the empirical distribution of the d_is: $P(d \geq v) = \#\{k | d_k \geq v\}/K$.
9. Compute the p-value: $\epsilon_0 = P(d \geq d_0)$.
10. Reject the null hypothesis that the two samples come from the same population if $\epsilon_0 < \epsilon$.

The above procedure, which is illustrated in Figs. 3 and 4, computes the null distribution of the distance function $\rho(X, Y)$ nonparametrically. In a standard parametric hypothesis-testing procedure, the forms of the distributions of x and y are known (usually assumed to be Gaussian) and, so, the null distribution of the test statistic $\rho(X, Y)$ is known. In contrast, the permutation test does not make any prior assumption regarding the distributions of x and y. Instead, an empirical null distribution is created by randomly permuting the data set Z and creating a histogram of computed test statistics (d_is).

By design, the size of the test, ϵ, is fixed. Thus, irrespective of the distance function $\rho(X, Y)$, the percentage of time that the validation procedure rejects a true null hypothesis that the two samples are from the same underlying population is ϵ. In other words, the probability of misdetection is ϵ. What is not fixed is the probability of false alarm, γ. which is the probability that the procedure claims that X and Y come from the same underlying population when, in fact, they come from different underlying populations. Although the use of various distance functions for ρ and δ gives rise to the same probability of misdetection ϵ, each has a different probability of false alarm γ. It is important to note that if two samples X and Y pass the validation procedure, this does not mean that we accept the null hypothesis. Rather, it means that we do not have enough statistical evidence to reject the null hypothesis. When we reject a null hypothesis, however, this *does* mean that we have enough statistical evidence to reject it.

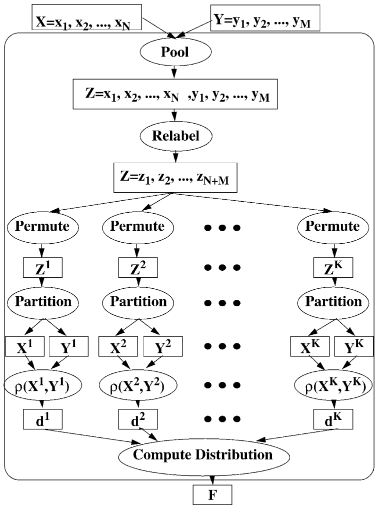


Fig. 4. This figure shows the permutation procedure for computing the null distributions.

4.3 Power Functions

Let us assume that the x_is are distributed as $F(\theta_X)$ and the y_is are distributed as $F(\theta_Y)$, where θ_X and θ_Y are the parameters of the distributions. Let the null hypothesis H_N and the alternate hypothesis H_A be

$$H_N : \quad \theta_X = \theta_Y \tag{1}$$

$$H_A : \quad \theta_X \neq \theta_Y. \tag{2}$$

The size of the test, ϵ, is fixed by the algorithm designer and is given as

$$\epsilon = P(H_A | H_N \text{ is true}). \tag{3}$$

The plot of 1 minus the probability of false alarm as a function of θ is the *power function (see Fig. 5).* Thus, if we fix the distribution parameter of the x_is at $\theta_X = \theta_0$, and vary the distribution parameter value $\theta_Y = \theta$ for the y_is, the power function is denoted by $\gamma_{\theta_0}(\theta)$, and is given by

$$\gamma_{\theta_0}(\theta) = P(H_A | \theta_X = \theta_0 \text{ and } \theta_Y = \theta) \ . \tag{4}$$

Thus, $1 - \gamma_{\theta_0}(\theta)$ is the probability of false alarm. The power function should have a minimum at $\theta_X = \theta_Y = \theta_0$, with $\gamma_{\theta_0}(\theta_0) = \epsilon$ and should increase on either side and go up to 1 when $\theta_Y = \theta$ is very far from θ_0.

Let us say there are two validation schemes A and B with test size ϵ and power functions $\gamma^A_{\theta_0}(\theta)$ and $\gamma^B_{\theta_0}(\theta)$. Since the misdetection probability ϵ is the same for both schemes, A is better than B if the false alarm rate of A is less than the false alarm rate of B. That is, A is better than B if $1 - \gamma^A_{\theta_0}(\theta) < 1 - \gamma^B_{\theta_0}(\theta)$ or $\gamma^A_{\theta_0}(\theta) > \gamma^B_{\theta_0}(\theta)$. If this relation is true for all values of θ, the procedure A is said to be uniformly more powerful than B. That is, the scheme A is better than scheme B if the power function plot of A is above the power function plot of B for all values of θ. Generalizing, if there are many validation schemes, the one whose power function is above all other power functions is the best scheme. If the power functions intersect, there is no clear winner; for some regions in the parameter space, one scheme is better while in other regions, the other scheme is better.

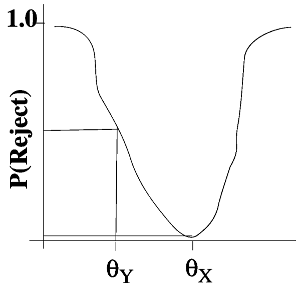


Fig. 5. The true parameter of the sample X is Θ_X. The parameter Θ_Y of the sample Y is updated and the corresponding probability of the test rejecting the null hypothesis that X and Y are from the same underlying distribution is plotted. The resulting curve is the power function.

For a given validation scheme, if we increase the sample sizes N and M, the power function changes and the new power function is higher than the old power function, and so by definition is more powerful. Thus, the sensitivity, i.e., the width of the notch at the minimum, is a function of the sample sizes N and M. When the sample size is small, the notch is broader, and when the sample size is large, the notch is sharper. This fact is used in deciding what sample size should be used for the test: Choose the sample size such that the desired probability of false alarm is attained when the parameters θ_X and θ_Y differ by a specified amount $\Delta\theta$.

Finally, our validation scheme described in the previous section is dependent on two distance functions ρ and δ. Thus, each choice of ρ and δ gives rise to a different power function. The combination that produces the highest power function is the best choice. See [1] for details on power functions.

4.4 Distance Functions, Outliers, and Robust Statistics

Various distance functions $\rho(X, Y)$ can be used for computing the distance between the sets of characters X and Y. We use the following symmetric distance functions for ρ (see Fig. 6).

Mean Nearest-Neighbor Distance:

$$\rho(X,Y) = \rho_{Mean}(X,Y) \;=\; \frac{\rho_{Mean}(Y;X) + \rho_{Mean}(X;Y)}{N+M},$$

where

$$\begin{aligned}\rho_{Mean}(Y;X) &= \sum_{x\in X}\left(\min_{y\in Y}\delta(x,y)\right)\\ \rho_{Mean}(X;Y) &= \sum_{y\in Y}\left(\min_{x\in X}\delta(x,y)\right).\end{aligned}$$

Trimmed Mean Nearest-Neighbor Distance:

$$\rho(X,Y) = \rho_{Trim}(X,Y) \;=\; \frac{\rho_{Trim}(Y;X) + \rho_{Trim}(X;Y)}{2},$$

where

$$\begin{aligned}\rho_{Trim}(Y;X) &= \mathrm{Trim}_{x\in X}\left(\min_{y\in Y}\delta(x,y)\right)\\ \rho_{Trim}(X;Y) &= \mathrm{Trim}_{y\in Y}\left(\min_{x\in X}\delta(x,y)\right).\end{aligned}$$

Here, the *Trim* function accepts as input a set of real numbers, orders them, and then discards the top and bottom 10 percent and returns the mean of the remaining 80 percent.

Median Nearest-Neighbor Distance:

$$\rho(X,Y) = \rho_{Med}(X,Y) \;=\; (\rho_{Med}(Y;X) + \rho_{Med}(X;Y))/2,$$

where

$$\begin{aligned}\rho_{Med}(Y;X) &= \mathrm{Median}\left(\min_{y\in Y}\delta(x,y)\right)\\ \rho_{Med}(X;Y) &= \mathrm{Median}\left(\min_{x\in X}\delta(x,y)\right).\end{aligned}$$

Notice that the mean nearest-neighbor distance is not a robust distance measure. That is, if for some reason a data point is far from the norm, the p-value computation becomes very sensitive to this data point. This can occur, for example, when a character in the real data set X is actually a "c" (instead of being an "e"), and is identified incorrectly as an "e." Yet, another outlier source is connected characters: when characters are extracted from a real document pieces of neighboring characters might get included in the bounding box of the extracted character. The median and the trimmed mean distance measures are robust against outliers since they do not look at the tails of the distribution. One would expect that these measures should work better in cases where there are outliers.

The distance function $\delta(x, y)$ mentioned earlier is the distance between two individual characters x and y. We use the Hamming distance for δ. This is computed by counting the number of pixels where the characters x and y differ after the centroids of x and y have been registered. A variety of other character distances $\delta(x, y)$ and set distance functions $\rho(X, Y)$ could have been used (e.g., the Hausdorff distance, rank-ordered Hausdorff distance, etc.). The combination of character distance $\delta(x, y)$ and set distance $\rho(X, Y)$ that give rise to the best power function is the best pair of character and set distances to use for the validation procedure.

5 Null Distribution for Gaussian Populations

In this section, we compute the null distributions of two set distances $\rho(X, Y)$ when x_i and y_i are Gaussian distributed. We show that when they are each Gaussian distributed with a known variance σ^2, the two distance functions considered are χ^2 distributed under the null hypothesis. Such closed-form solutions for the null distributions are possible only when the underlying distributions are known a priori. However, this is not the case, in general—the Gaussian assumptions might be appropriate in some settings but completely wrong in other settings. Thus, the nonparametric permutation method described in Section 4 is a much better approach to computing the null distributions when the forms of the sample distributions are not known. Nevertheless, for the purpose of validating the software and algorithm for computing the empirical null distribution, the Gaussian case is very useful since it allows us to compare the empirical distributions against known (theoretically computed) distributions.

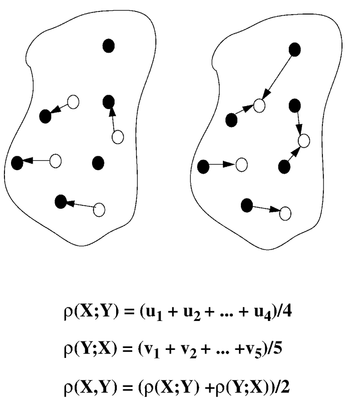

ρ(X;Y) = (u1 + u2 + ... + u4)/4
ρ(Y;X) = (v1 + v2 + ... +v5)/5
ρ(X,Y) = (ρ(X;Y) +ρ(Y;X))/2

Fig. 6. The black dots are elements of the set X and the white dots are elements of the set Y. In the figure on the left, the distance $\rho(X;Y)$ from Y to X is computed by summing the distance of each y_i to the nearest x_i. Similarly on the right the distance $\rho(Y;X)$ is computed. The final symmetric distance $\rho(X,Y)$ is computed by taking the mean.

5.1 Intercluster Mean Distance

Let $X = \{x_1, x_2, \ldots, x_N\}$ be a set such that $x_i \in R$ and $x_i \sim N(\mu_X, \sigma^2)$. Similarly, let $Y = \{y_1, y_2, \ldots, y_N\}$ be a set such that $y_i \in R$ and $y_i \sim N(\mu_Y, \sigma^2)$. The problem is to test the null hypothesis that $\mu_X = \mu_Y$ when σ^2 is known.

Now, we know that

$$\hat{\mu}_X = \frac{1}{N}\sum_{i=1}^{N} x_i \sim N(\mu_X, \sigma^2/N) \tag{5}$$

$$\hat{\mu}_Y = \frac{1}{N}\sum_{i=1}^{N} y_i \sim N(\mu_Y, \sigma^2/N). \tag{6}$$

Therefore,

$$\hat{\mu}_X - \hat{\mu}_Y \sim N(\mu_X - \mu_Y, 2\sigma^2/N) \tag{7}$$

and

$$\sqrt{N/2}(\hat{\mu}_X - \hat{\mu}_Y)/\sigma \sim N(\mu_X - \mu_Y, 1)\ . \tag{8}$$

Now, let

$$t = \rho(X,Y) = \frac{N}{2\sigma^2}(\hat{\mu}_X - \hat{\mu}_Y)^2.$$

Thus, under the null hypothesis that $\mu_X = \mu_Y$, we have

$$t = \rho(X,Y) \sim \chi_1^2. \tag{9}$$

Thus, instead of empirically computing the distributions as described in Section 4, we can use the above analytic form of the distribution to accept or reject the null hypothesis. Moreover, we see that the empirical method has reduced to a standard statistical technique when the underlying distribution is known to be Gaussian.

5.2 Likelihood Distance

In the previous section, we picked a particular distance function $\rho(X,Y)$ and showed that its null distribution is χ_1^2. In this section, we pick a distance function based on the likelihood function of the data. It turns out that this distance function is the same as the one used in the previous section.

Let $X = \{x_1, x_2, \ldots x_N\}$, where $x_i \in R$ and $x_i \sim N(\mu_X, \sigma^2)$. Similarly, let $Y = \{y_1, y_2, \ldots, y_N\}$, where $y_i \in R$, and $y_i \sim N(\mu_Y, \sigma^2)$. The problem is to test the null hypothesis that $\mu_X = \mu_Y = \mu$.

Let $\rho_Y(X)$ denote the distance of set X from set Y. Here, we use a function of the likelihood for ρ

$$\rho_X(Y) = f(P(y_1, \ldots, y_N | x_1, \ldots, x_N, \sigma)) \tag{10}$$

$$\rho_Y(X) = f(P(x_1, \ldots, x_N | y_1, \ldots, y_N, \sigma)). \tag{11}$$

In general, the above distances need not be symmetric in X and Y. Hence, we also consider symmetric distances of the form

$$\rho(X,Y) = \\ f(P(y_1, \ldots, y_N | x_1, \ldots, x_N, \sigma) P(x_1, \ldots, x_N | y_1, \ldots, y_N, \sigma)). \tag{12}$$

Also, we can consider the right hand side in the equation above divided by $\log \max_\mu P(x_1, \ldots, x_N, y_1, \ldots, y_N | \mu, \sigma)$. That is,

$$\rho(X,Y) = \\ \log\left(\frac{P(y_1, \ldots, y_N | x_1, \ldots, x_N, \sigma) P(x_1, \ldots, x_N | y_1, \ldots, y_N, \sigma)}{\max_\mu P(x_1, \ldots, x_N, y_1, \ldots, y_N | \mu, \sigma)}\right). \tag{13}$$

We can use the standard rules of probability theory to manipulate the above equation as follows:

$$\begin{aligned} & P(y_1, \ldots, y_N | x_1, \ldots, x_N, \sigma) \\ &= \int_{-\infty}^{\infty} P(\mu, y_1, \ldots, y_N | x_1, \ldots, x_N, \sigma) d\mu \\ &= \int_{-\infty}^{\infty} \frac{P(y_1, \ldots, y_N, x_1, \ldots, x_N, \mu, \sigma)}{P(x_1, \ldots, x_N, \sigma)} d\mu \\ &= \int_{-\infty}^{\infty} \frac{P(y_1, \ldots, y_N | x_1, \ldots, x_N, \mu, \sigma) P(x_1, \ldots, x_N, \mu, \sigma)}{P(x_1, \ldots, x_N, \sigma)} d\mu \\ &= \int_{-\infty}^{\infty} \frac{P(y_1, \ldots, y_N | \mu, \sigma) P(x_1, \ldots, x_N | \mu, \sigma) P(\mu, \sigma)}{\int_{-\infty}^{\infty} P(x_1, \ldots, x_N | \lambda, \sigma) P(\lambda, \sigma) d\lambda} d\mu. \end{aligned} \tag{14}$$

Now, we make the assumption that μ and σ are independent so that $P(\mu, \sigma) = P(\mu)P(\sigma)$. Furthermore, we assume that μ and σ have a uniform prior. Although this implies the prior is improper (since its integral is not equal to 1), the posterior distribution integrates to 1. Thus, $P(\mu, \sigma) = P(\mu)P(\sigma) \propto \epsilon$. But the ϵ in the numerator and the denominator of (14) cancel out and the numerator can now be written as follows:

$$\begin{aligned} & P(y_1, \ldots, y_N | \mu, \sigma) P(x_1, \ldots, x_N | \mu, \sigma) P(\mu, \sigma) \\ &= \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^N e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - \mu)^2} \cdot \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^N e^{-\frac{1}{2\sigma^2}\sum_{j=1}^{N}(x_j - \mu)^2} \\ &= \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^{2N} e^{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^{N}(y_i - \mu)^2 + \sum_{j=1}^{N}(x_j - \mu)^2\right]}. \end{aligned} \tag{15}$$

Since the denominator is not a function of either μ or $y_1, \ldots, y_N$, it is a constant. The denominator can be computed by integrating out $\mu, y_1, \ldots, y_N$ from the probability density in (15). Thus,

$$\begin{aligned} &P(y_1, \ldots, y_N | x_1, \ldots, x_N, \sigma) \\ &= C \int_{-\infty}^{\infty} \left(\frac{1}{\sqrt{2\pi}\sigma} \right)^{2N} e^{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^{N}(y_i-\mu)^2 + \sum_{j=1}^{N}(x_j-\mu)^2\right]}, \end{aligned} \tag{16}$$

where the constant of integration C can be found by equating the right hand side to 1. In order to compute the integral, we simplify the exponent inside the integral:

$$\begin{aligned} &\sum_{i=1}^{N}(y_i-\mu)^2 + \sum_{j=1}^{N}(x_j-\mu)^2 \\ &= \sum_{i=1}^{N}(y_i - \bar{y} + \bar{y} - \mu)^2 + \sum_{j=1}^{N}(x_i - \bar{x} + \bar{x} - \mu)^2 \\ &= \sum_{i=1}^{N}(y_i - \bar{y})^2 + \sum_{i=1}^{N}(y_i - \bar{y})(\bar{y} - \mu) + N(\bar{y} - \mu)^2 \\ &\quad \sum_{j=1}^{N}(x_i - \bar{x})^2 + \sum_{j=1}^{N}(x_i - \bar{x})(\bar{x} - \mu) + N(\bar{x} - \mu)^2 \\ &= \sum_{i=1}^{N}(y_i - \bar{y})^2 + \sum_{j=1}^{N}(x_i - \bar{x})^2 + N(\bar{y} - \mu)^2 + N(\bar{x} - \mu)^2. \end{aligned} \tag{17}$$

But,

$$\begin{aligned} (\bar{y} - \mu)^2 + (\bar{x} - \mu)^2 &= \bar{x}^2 + \bar{y}^2 + 2\left[\mu^2 - 2\mu\left(\frac{\bar{x} + \bar{y}}{2}\right)\right] \\ &= \bar{x}^2 + \bar{y}^2 - 2\mu\left(\frac{\bar{x} + \bar{y}}{2}\right)^2 \\ &\quad + 2\left[\mu^2 - 2\mu\left(\frac{\bar{x} + \bar{y}}{2}\right) + \left(\frac{\bar{x} + \bar{y}}{2}\right)^2\right] \\ &= \frac{(\bar{x}^2 + \bar{y}^2 - 2\bar{x}\bar{y})}{2} + 2\left[\mu - \left(\frac{\bar{x} + \bar{y}}{2}\right)\right]^2 \\ &= \frac{(\bar{x} - \bar{y})^2}{2} + 2\left[\mu - \left(\frac{\bar{x} + \bar{y}}{2}\right)\right]^2. \end{aligned} \tag{18}$$

Thus, from (17) and (18)

$$\begin{aligned} &\sum_{i=1}^{N}(y_i-\mu)^2 + \sum_{j=1}^{N}(x_j-\mu)^2 \\ &= \sum_{i=1}^{N}(x_i - \bar{x})^2 + \sum_{j=1}^{N}(y_j - \bar{y})^2 \\ &\quad + \frac{N}{2}(\bar{x} - \bar{y})^2 + 2N\left(\mu - \left(\frac{\bar{x} + \bar{y}}{2}\right)\right)^2. \end{aligned} \tag{19}$$

Also, since

$$\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}(\sigma/\sqrt{2N})} e^{-\frac{1}{2\sigma^2/2N}\left(\mu - \frac{\bar{x}+\bar{y}}{2}\right)^2} d\mu = 1, \tag{20}$$

we have

$$\begin{aligned} &P(y_1, \ldots, y_N | x_1, \ldots, x_N, \sigma) \\ &= C\left(\frac{1}{\sqrt{2\pi}\sigma}\right)^{2N} \frac{\sqrt{2\pi}\sigma}{\sqrt{2N}} \cdot e^{\frac{1}{2\sigma^2/2N}\left[\sum_{i=1}^{N}(x_i-\bar{x})^2 + \sum_{j=1}^{N}(y_j-\bar{y})^2 + \frac{N}{2}(\bar{x}-\bar{y})^2\right]}. \end{aligned} \tag{21}$$

Now, to get the value of C, we proceed as follows:

$$\begin{aligned} 1 &= C \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \\ &\quad \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^{2N} e^{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^{N}(y_i-\mu)^2 + \sum_{j=1}^{N}(x_j-\mu)^2\right]} dy_1 \ldots dy_N d\mu \\ &= C \int \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^{N} e^{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^{N} N(x_i-\bar{x})^2 + N(\mu-\bar{x})^2\right]} d\mu \\ &= C\left(\frac{1}{\sqrt{2\pi}\sigma}\right)^{N} \frac{\sqrt{2\pi}\sigma}{\sqrt{N}} e^{-\frac{1}{2\sigma^2/N}\left[\sum_{i=1}^{N}(x_i-\bar{x})^2\right]}. \end{aligned} \tag{22}$$

Thus, we have computed C to be

$$C = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^{-(N+1)} \sqrt{N} e^{\frac{1}{2\sigma^2/N}\left[\sum_{i=1}^{N}(x_i-\bar{x})^2\right]}. \tag{23}$$

Now, we can write the complete conditional density as

$$\begin{aligned} &P(y_1, \ldots, y_N | x_1, \ldots, x_N, \sigma) \\ &= \left[\left(\frac{1}{\sqrt{2\pi}\sigma}\right)^{-(N+1)} \sqrt{N} e^{\frac{1}{2\sigma^2/N}\left[\sum_{i=1}^{N}(x_i-\bar{x})^2\right]}\right] \\ &\quad \cdot \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^{2N} \frac{\sqrt{2\pi}\sigma}{\sqrt{2N}} \cdot e^{\frac{1}{2\sigma^2/2N}\left[\sum_{i=1}^{N}(x_i-\bar{x})^2 + \sum_{j=1}^{N}(y_j-\bar{y})^2 + \frac{N}{2}(\bar{x}-\bar{y})^2\right]} \\ &= \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^{N} \cdot \sqrt{2} \cdot e^{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^{N}(y_i-\bar{y})^2 + \frac{N}{2}(\bar{x}-\bar{y})^2\right]}. \end{aligned} \tag{24}$$

Thus, we can use $2\sigma^2$ times the negative exponent of the conditional probability, as given in (24), as the test statistic $\rho_X(Y)$. Notice that it is not symmetric in X and Y.

$$\rho_X(Y) = f(P(y_1, \ldots, y_N | x_1, \ldots, x_N, \sigma)) \tag{25}$$

$$\begin{aligned} &= -\log P(y_1, \ldots, y_N | x_1, \ldots, x_N, \sigma) \\ &\quad + \frac{N}{2}\log(2\pi\sigma^2) - \frac{1}{2}\log(2) \end{aligned} \tag{26}$$

$$= \sum_{i=1}^{N}(y_i - \bar{y})^2 + \frac{N}{2}(\bar{x} - \bar{y})^2. \tag{27}$$

$$\rho_Y(X) = f(P(x_1, \ldots, x_N | y_1, \ldots, y_N, \sigma)) \tag{28}$$

$$\begin{aligned} &= -\log P(x_1, \ldots, x_N | y_1, \ldots, y_N, \sigma) \\ &\quad + \frac{N}{2}\log(2\pi\sigma^2) - \frac{1}{2}\log(2) \end{aligned} \tag{29}$$

$$= \sum_{i=1}^{N}(x_i - \bar{x})^2 + \frac{N}{2}(\bar{y} - \bar{x})^2. \tag{30}$$

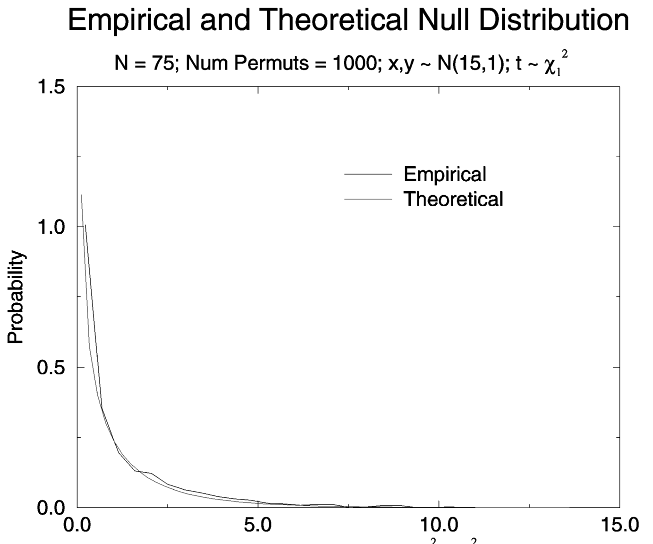


Fig. 7. Empirical and theoretical null distributions for two sample tests. Samples X and Y of size $N = 75$ are drawn from $N(15,1)$. The empirical null distribution is computed as described in Section 4. We use 1,000 random permutations for computing the distribution. The distance function used is $t = \rho(X,Y) = N(\bar{x} - \bar{y})^2/(2\sigma^2)$. The theoretical distribution of t is χ_1^2. The empirical and theoretical plots have been plotted together in this figure.

In order to get a symmetric test statistic, we can look at the product of the conditional probabilities, so that

$$\begin{aligned}&\rho_X(Y) + \rho_Y(X)\\ &= \sum_{i=1}^{N}(y_i - \bar{y})^2 + \frac{N}{2}(\bar{x} - \bar{y})^2 + \sum_{i=1}^{N}(x_i - \bar{x})^2 + \frac{N}{2}(\bar{y} - \bar{x})^2.\end{aligned} \tag{31}$$

But, we know that the sum of the within-cluster scatter and the between-cluster scatter is equal to the total scatter. Thus,

$$\begin{aligned}&\sum_{i=1}^{N}(y_i - \bar{y})^2 + \frac{N}{2}(\bar{x} - \bar{y})^2 + \sum_{i=1}^{N}(x_i - \bar{x})^2\\ &= \sum_{i=1}^{N}\left(x_i - \left(\frac{\bar{x} + \bar{y}}{2}\right)\right)^2 + \sum_{j=1}^{N}\left(y_j - \left(\frac{\bar{x} + \bar{y}}{2}\right)\right)^2.\end{aligned}$$

Notice that for given data sets, the above summation is the same constant regardless of which points go with x_i and which with y_i. Thus,

$$\rho_X(Y) + \rho_Y(X) = C + \frac{N}{2}(\bar{y} - \bar{x})^2, \tag{32}$$

where C is a constant. Thus, a symmetric test statistic based on likelihood is

$$\rho(X,Y) = \frac{N}{2\sigma^2}(\bar{y} - \bar{x})^2. \tag{33}$$

The reason for normalizing by σ^2 will become clear shortly.

The Monte Carlo hypothesis tests can now be conducted with the distance functions ρ defined in this section. In Fig. 7, we show that the theoretically computed null distribution agrees with the null distribution computed empirically by random permutations.

It is important to statistically compare the test statistics $\rho_X(Y), \rho_Y(X),$ and $\rho(X,Y)$ computed in this section. Notice that

$$\begin{aligned}\bar{x} &\sim N(0, \sigma^2/N),\\ \bar{y} &\sim N(0, \sigma^2/N),\\ \bar{x} - \bar{y} &\sim N(0, 2\sigma^2/N).\end{aligned}$$

Thus,

$$(\bar{x} - \bar{y})^2 \sim \frac{2\sigma^2}{N}\chi_1^2$$

and

$$\rho(X,Y) = \frac{N}{2\sigma^2}(\bar{x} - \bar{y})^2 \sim \chi_1^2. \tag{34}$$

Thus, $\rho(X,Y)$ has a mean of 1 and variance of 2. Similarly,

$$\frac{1}{\sigma^2}\sum_{i=1}^{N}(y_i - \bar{y})^2 \sim \chi_{N-1}^2.$$

Thus,

$$\frac{1}{\sigma^2}\sum_{i=1}^{N}(y_i - \bar{y})^2 + \frac{N}{2\sigma^2}(\bar{x} - \bar{y})^2 \sim \chi_{N-1}^2 + \chi_1^2,$$

so that

$$\rho_X(Y) \sim \chi_N^2. \tag{35}$$

We see that $\rho_X(Y)$ has a mean of N and variance of $2N$. This implies that $\rho(X,Y)$ is a more powerful test statistic (in terms of false alarms) than $\rho_X(Y)$ or $\rho_Y(X)$.

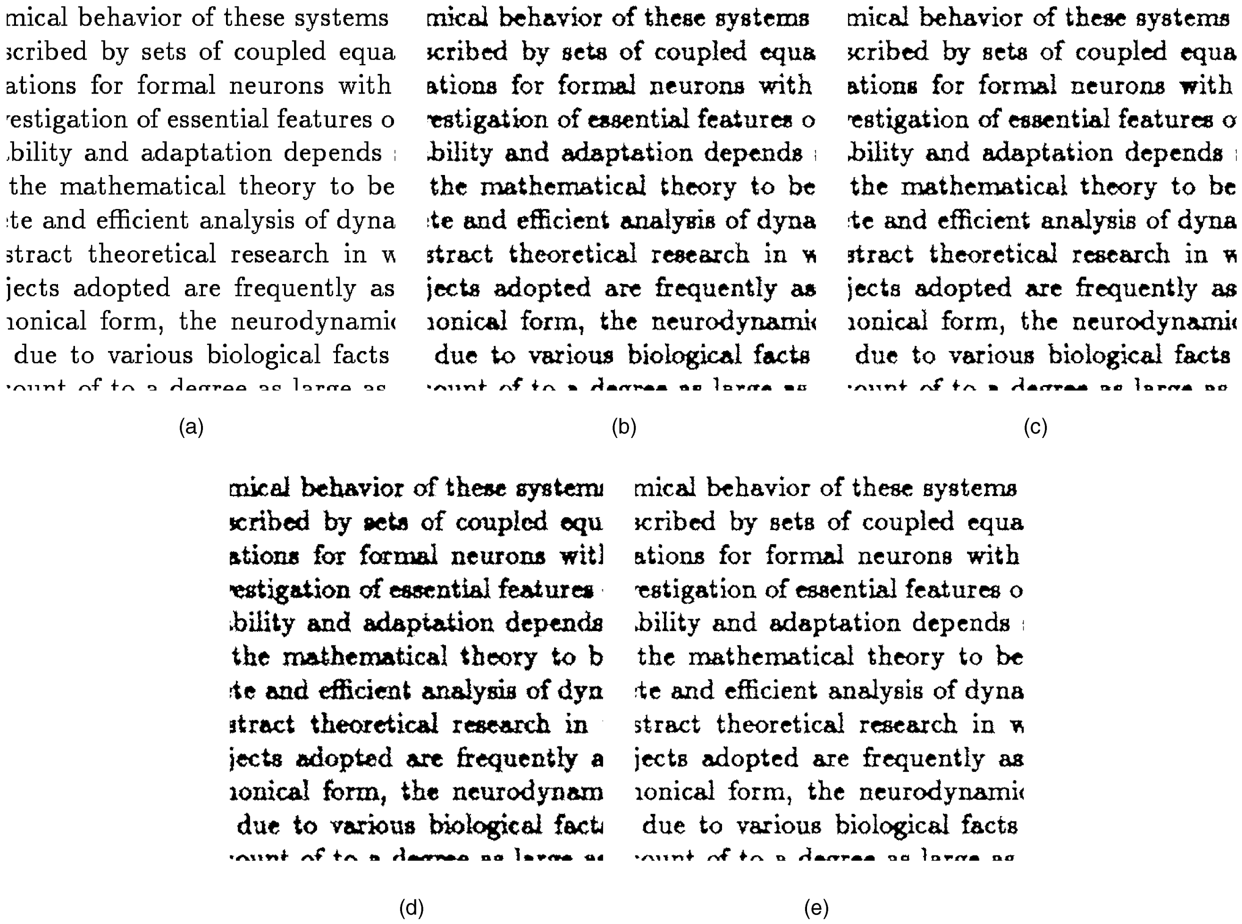
6 Experimental Protocol and Results

In this section, we outline the protocol we use to conduct the experiments. Here, we give all the sample sizes we use, the number of trials that are run at different stages, the exact model parameter values that are used for generating the synthetically degraded characters, the impact of the distance functions, etc. Three types of experiments are possible:

Synthetic vs. Synthetic: One sample X is synthetically created using the document degradation model, with a fixed model parameter value. Then, many samples Y are generated, again using the model, but with different parameter settings. The validation procedure can be run on the samples X and Y and the power function generated. This experiment is in part a sanity check for the methodology: If it does not work on controlled synthetic data, there is little point in trying it on real data.

Real vs. Real: This experiment tests for systematic dissimilarities between two image populations (e.g., rotations, fonts, etc.). Note that this use of the validation procedure is independent of the degradation model.

Real vs. Synthetic: Here, the sample X consists of real degraded characters and the sample Y is generated by varying the degradation model parameter Θ. The validation procedure is run on the X and Y samples



mical behavior of these systems
scribed by sets of coupled equa
ations for formal neurons with
estigation of essential features o
bility and adaptation depends
the mathematical theory to be
te and efficient analysis of dyna
stract theoretical research in w
jects adopted are frequently as
ionical form, the neurodynamic
due to various biological facts

Fig. 8. Local document degradation model. (a) Subimage of the noise-free document. (b) Degraded document generated with $\alpha = \beta = 1.5$. (c) A degraded image accepted as similar to (b), $\alpha = \beta = 1.7$, (d) A degraded image rejected as similar to (b), $\alpha = \beta = 0.9$. (e) A degraded image rejected as similar to (b), $\alpha = \beta = 2.0$. The sample size used is 60.

and a power function is generated. This experiment tests whether or not the synthetic characters are actually close to the real characters.

6.1 Protocol for Synthetic vs. Synthetic

The following protocol is used for creating the samples X and Y. The distribution parameter Θ_X is fixed with the following parameter component values: $\eta_f = \eta_b = 0$, $\alpha_0 = \beta_0 = 1$, and $\alpha = \beta = 1.5$ and structuring element size $k = 5$. The distribution parameter Θ_Y is varied by varying α and β. In our experiments, we make α equal to β. The other parameter components of Θ_Y—$\eta_f, \eta_b, \alpha_0, \beta_0, k$—are made equal to the corresponding components of the model parameter Θ_X. In all cases, the noise-free document is the same (a LaTeX document page formatted in IEEE Transactions style) and the same set of 340 "e" characters (Computer Modern Roman 10 point font) are extracted from the page to create the samples X and Y.

The validation procedure parameters used are as follows:

1. Sizes of samples X and Y: $N = M = \{10, 20, 60\}$.
2. Number of permutations: $K = 1,000$.
3. Significance level of the test: $\epsilon = 0.05$.
4. Number of repetitions used in computing the power function: $T = 100$.
5. The character-to-character distance $\delta(x, y)$ used is the Hamming distance.
6. The set-to-set distance $\rho(X, Y)$ used is the mean nearest-neighbor distance.

The noise-free document is shown in Fig. 8a. The degraded document generated with model parameter Θ_X is shown in Fig. 8b. The power functions for sample sizes 10, 20, 60 are shown in Fig. 9. The power function corresponding to sample size 10 is the widest and the power function corresponding to sample size 60 is the narrowest. Note that all three power functions give a misdetection (reject) rate close to $\epsilon = 0.05$ when Θ_Y is close to Θ_X. (Only the α component, which is equal to 1.5 for Θ_X, is shown in the plot.) Furthermore, when the α component for Θ_Y is far from 1.5, the misdetection rate is close to 1.0, which implies that the validation procedure can distinguish the two samples with high probability. An image generated with $\alpha = \beta = 1.7$ that the validation procedure accepted with a probability close to 0.9 is shown in Fig. 8c. Two document images generated with parameter values $\alpha = \beta = 2.0$ and $\alpha = \beta = 0.9$ that are easily rejected by the validation procedure are shown in Fig. 8d and Fig. 8e, respectively.

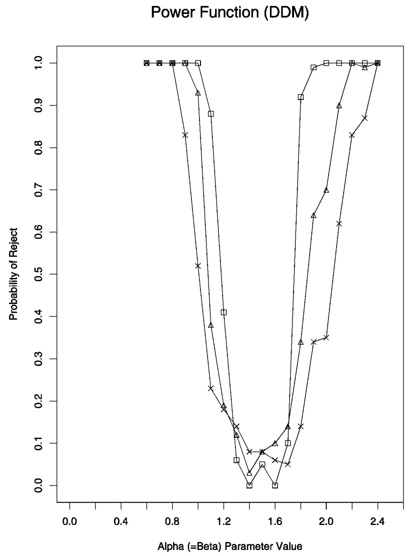


Fig. 9. Power plots for the local document degradation model. The parameters for X were fixed with $\alpha = \beta = 1.5$, while the parameters for Y were varied. Notice that the power function has a minimum near $\alpha = \beta = 1.5$. The power function corresponding to a sample size of 60 (boxes) is sharper; that corresponding to a sample size of 10 (crosses) is broader.

6.2 Protocol for Real vs. Real Experiment

First, various European language texts are generated using the Adobe Times-Roman typeface at 8 point. Next, these documents are printed on a Canon laser printer and then scanned at 400 pixels per inch using a Canon scanner. Lower-case "e"s are extracted semiautomatically by OCR (thus, some characters possess artifacts resulting from resegmentation). From among these, 3,000 characters are selected by two persons working independently to avoid misclassifications.

Before selecting the two populations, we randomly shuffle the real data in order to obscure any systematic per-page dissimilarities (due to, for example, skew scale variations). The validation procedure does not reject the null hypothesis that the two samples are from the same underlying population. Repeated trials give a reject rate close to 0.05, the significance level designed into the test.

6.3 Outliers and Distance Function Comparisons

The validation procedure protocol is as follows: The significance level ϵ is fixed at 0.05, the sample sizes $N = M$ used are 10, 20, and 60, the number of permutations K for creating the empirical null distribution is 1,000, and the number of trials T for estimating the misdetection rate is 100.

We studied the sensitivity of the validation procedure to the set distance $\rho(X, Y)$ as follows: The data sets X and Y are collections of (synthetic) degraded characters "e." Degradation parameter values for X are fixed at $\alpha = \beta = 1.5$, but the corresponding degradation parameters for Y are varied from 0.6 to 2.4. The Hamming distance is used for the character-to-character distance $\delta(x, y)$. The sample size of X and Y is fixed at $N = M = 60$. The mean, trimmed mean, and median distances are used to compute the power function, in both the presence and absence of outliers.

Figs. 10a, 11a, and 12a show the power functions in the absence of outliers when the mean and the trimmed mean distances are used. Next, we introduced outliers into the data set X by replacing five degraded "e"s with degraded "c"s. The Y data set is unchanged. Figs. 10b, 11b, and 12b, show the power functions in the presence of outliers. Clearly, the median and trimmed mean nearest-neighbor distances are more robust against outliers since the corresponding power functions are not affected. Furthermore, it can be seen that the median NN distance function, in the outlier-free case, is less "powerful" than the mean NN distance function since the median distance power function lies below the mean distance power function plot. Finally, it can be seen that the 10 percent trimmed NN distance function is superior to the other two distance functions, since the corresponding power function is robust against outliers and at the same time higher.

6.4 Protocol for Validating Real vs. Synthetic Degradations

The ideal data is a LaTeX formatted document. The IEEE Transactions style is used for typesetting the document. The corresponding ideal binary image and character groundtruth are created using the DVI2TIFF software. The ideal document is created at 300×300 dots/inch resolution and the size of the binary document in pixels is $3,300 \times 2,550$. This document is printed using a SparcPrinter II. Next, the original printed document is photocopied five times using a Xerox photocopier—once at the normal setting, twice with darker settings, and twice with lighter settings. Finally, the five photocopied documents are scanned using a Ricoh scanner. The scanner is set at 300×300 dots/inch resolution. The rest of the scanner parameters are set at normal settings. The scanned binary image is of size 3307×2544. The parameters are then estimated using the protocol specified in [8]. In all cases, the noise-free document is the same (a LaTeX document page formatted in IEEE Transactions style) and the same set of 340 characters "e" (Computer Modern Roman 10 point font) is extracted from the page to create the synthetic population Y.

The validation procedure parameters used are as follows:

1. Sample sizes of scanned characters X and synthetic characters Y: $N = M = \{10, 20, 60\}$.
2. Number of permutations for creating the empirical null distribution: $K = 1,000$.
3. Significance level of the test: $\epsilon = 0.05$.
4. Number of bootstrap repetitions for computing the reject rate of the test: $T = 100$.
5. The bootstrap samples are sampled (with replacement) from a pool of size $N_b = 100$.
6. The character-to-character distance $\delta(x, y)$ used is the Hamming distance.
7. The set-to-set distance $\rho(X, Y)$ used is the mean nearest-neighbor distance.

The above test was conducted on "e"s. The test did not reject the null hypothesis that the samples are from the same population for a sample size of 10. That is, the reject rate is lower than 5 percent. For the sample size of 20,

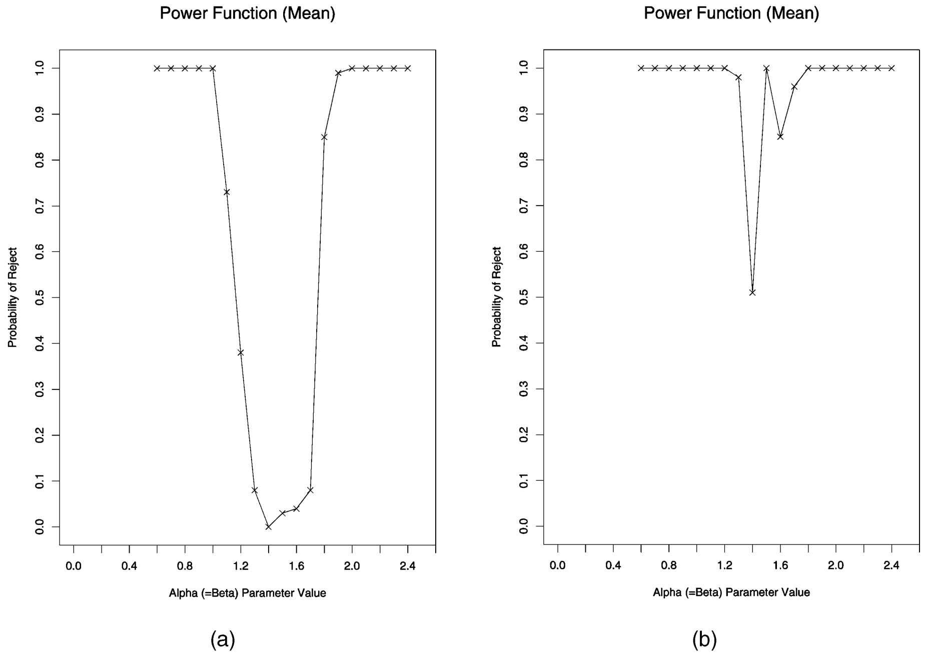


Fig. 10. Power functions of the validation procedure when mean, nearest-neighbor distance is used for the set distance function $\rho(X, Y)$. (a) When there are no outliers. (b) Corresponds to the situation when there are five outliers in one of the data sets.

46 percent of the time the test rejected the null hypothesis. For sample size of 60, the null hypothesis is rejected 100 percent of the time.

7 Comparing Two Models

In the previous section, we used a two-sample permutation procedure to test the null hypothesis that the sample of real degraded characters and the sample generated by the estimated degradation model are from the same underlying population. We found that when the sample size is 40, the test procedure rejects the null hypothesis.

In fact, in a two-sample test, if one of the samples is from a distribution that is even slightly different from the second sample's distribution, the statistical testing procedure will be able to reject the null hypothesis that the samples are from the same underlying population if the sample size is large enough.

Since we know that any model of a real process, with very high likelihood, is an approximation to the real

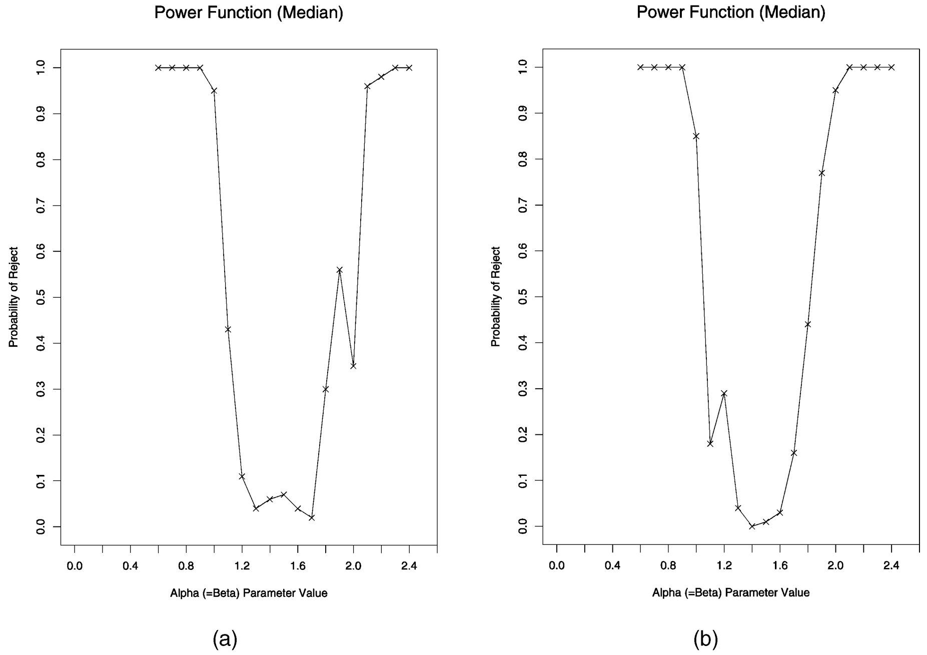


Fig. 11. Power functions of the validation procedure when median, nearest-neighbor distance is used for the set distance function $\rho(X, Y)$. (a) When there are no outliers. (b) Corresponds to the situation when there are five outliers in the X data set.

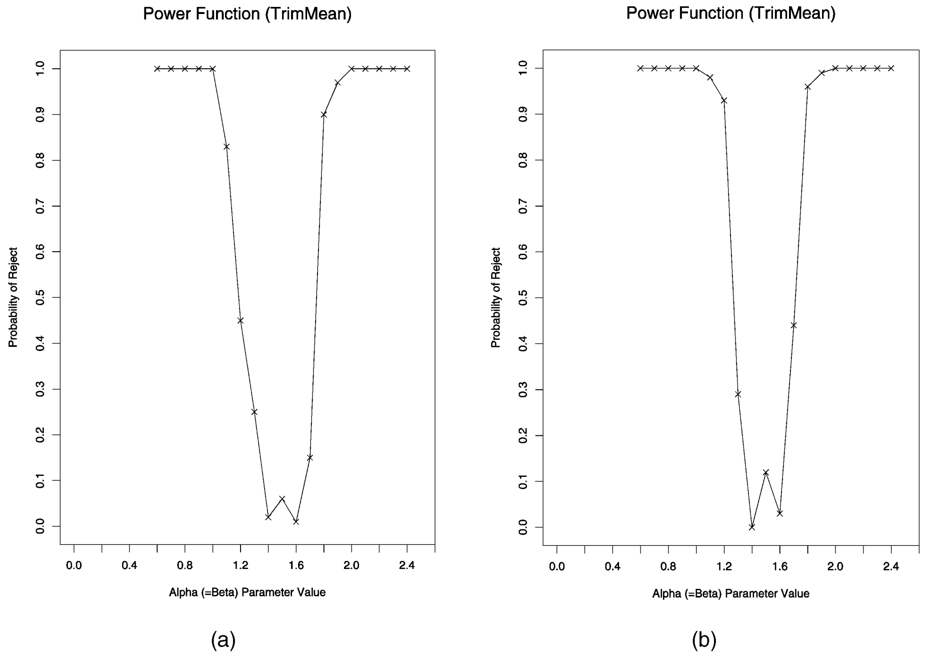


Fig. 12. Power functions of the validation procedure when 10 percent trimmed mean, nearest-neighbor distance is used for the set distance function $\rho(X, Y)$. (a) When there are no outliers. (b) Corresponds to the situation when there are five outliers in the X data set.

process, the samples generated from the model will be different from the real samples. Thus, any validation procedure will be able to distinguish the real and synthetic samples if the sample sizes are large enough. In other words, it is futile to test the equality of the distributions of the synthetic samples and the real samples; they will always be proved to be unequal if a large enough sample size is used. Even if some other validation procedure is used, for example, any method based on comparison of confusion matrices, the equality test is always going to give a negative result when the sample size is made large enough.

The next question is: How can one use the validation procedure in practice if the models are always going to be proved incorrect? The way to use the validation procedure is to compare two models and not evaluate just one model. That is, one can use the validation procedure to determine which model is closer to reality.

Let us say there are two document degradation models M_1 and M_2. The problem is to find the model that is closer to the real process. We know that if the sample size N of the synthetic and real samples is increased, after a certain point, the validation procedure will start rejecting both models. However, we will now give a procedure that will allow a researcher to decide which model is closer to reality for a fixed sample size N.

1. Fix the sample size N.
2. We are given a real sample D of size N.
3. Generate synthetic samples S_1 and S_2 of size N using the models M_1 and M_2, respectively.
4. Conduct the two-sample validation test using the real sample D and the synthetic sample S_1. Let the associated p-value be p_1.
5. Conduct the two-sample validation test using the real sample D and the synthetic sample S_2. Let the associated p-value be p_2.
6. If $p_1 > p_2$, model M_1 is closer to the real process for a sample size of N. Otherwise, model M_2 is closer.

Thus, the above procedure allows a researcher to choose between models. When we were choosing between parameter settings for a fixed model, we could use the power function to arrive at the best parameter sitting. However, two different models have *different* parameter spaces and, hence, they cannot compared using power functions. The p-value provides a means of comparing the models on a common basis.

8 Conclusions

We have posed the degradation model validation problem as a statistical, two-sample, hypothesis testing problem. A nonparametric permutation test is used for this purpose. The user specifies a test statistic, which is essentially a distance function on the two sets of degraded characters. The null distribution of the test statistic, which is the distribution of the test statistic under the hypothesis that the two samples come from the same underlying population, is created using a permutation procedure. The p-value corresponding to the test statistic associated with the two sets is computed and compared with a user-specified significance level to reject or not reject the null hypothesis. This procedure and several robust variants are implemented and evaluated empirically. The goodness of the distance functions is evaluated using power functions, which are standard statistical devices. The local degradation model passes the validation test when the sample size is small but rejects it when the sample size is increased. This is so because any model of a real-world process is an approximation and, thus, will not pass the test if the sample size is increased. Another way of using the validation procedure is for choosing between models. After the validation procedure is run, a p-value is obtained. Thus, if two different models are tested on the same real data, each validation procedure gives rise to a p-value for each model. The model whose associated p-value is larger is in closer agreement with the real data and thus should be preferred.

Acknowledgments

The authors would like to thank Azriel Rosenfeld of the University of Maryland for his comments. This research was funded in part by the US Department of Defense and the Army Research Laboratory under Contract MDA 9049-6C-1250.

References

[1] S.F. Arnold, *Math. Statistics.* N.J.: Prentice-Hall, 1990.

[2] H. Baird, "Document Image Defect Models," *Proc. IAPR Workshop Syntactic and Structural Pattern Recognition,* pp. 38-46, June 1990.

[3] H. Baird, "Calibration of Document Image Defect Models," *Proc. Second Ann. Symp. Document Analysis and Information Retrieval,* pp. 1-16, Apr. 1993.

[4] H.S. Baird, "Document Image Defect Models," *Structured Document Image Analysis.* New York: Springer-Verlag, 1992.

[5] G. Borgerfors, "Distance Transforms in Digital Images," *Computer Vision, Graphics, and Image Processing,* vol. 34, pp. 344-371, 1986.

[6] B. Efron and R.J. Tibshirani, *An Introduction to the Bootstrap.* New York: Chapman and Hall, 1993.

[7] P. Good, *Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses,* New York: Springer-Verlag, 1994.

[8] T. Kanungo, "Document Degradation Models and a Methodology for Degradation Model Validation," PhD thesis, Univ. of Washington, Seattle, 1996. http://www.cfar.umd.edu/~kanungo/pubs/phdthesis.ps.Z.

[9] T. Kanungo and R.M. Haralick, "Morphological Degradation Parameter Estimation," *Proc. SPIE Conf. Nonlinear Image Processing,* vol. 2,424, pp. 86-95, Feb. 1995.

[10] T. Kanungo, R.M. Haralick, H.S. Baird, W. Stuetzle, and D. Madigan, "Document Degradation Models: Parameter Estimation and Model Validation," *Proc. Int'l Workshop Machine Vision Applications,* pp. 552-557, Dec. 1994.

[11] T. Kanungo, R.M. Haralick, and I. Phillips, "Global and Local Document Degradation Models," *Proc. Second Int'l Conf. Document Analysis and Recognition,* pp. 730-734, Oct. 1993.

[12] T. Kanungo, R.M. Haralick, and I. Phillips, "Nonlinear Local and Global Document Degradation Models," *Int'l J. Imaging Systems and Technology,* vol. 5, pp. 220-230, 1994.

[13] D.E. Knuth, *TEX: The Program.* Mass.: Addison-Wesley, 1988.

[14] L. Lamport, *LATEX: A Document Preparation System,* Mass.: Addison-Wesley, 1986.

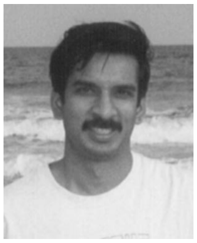
[15] Y. Li, D. Lopresti, and A. Tomkins, "Validation of Document Defect Models for Optical Character Recognition," *Proc. Third Ann. Symp. Document Analysis and Information Retrieval,"* pp. 137-150, Apr. 1994.

[16] Y. Li, D. Lopresti, and A. Tomkins, "Validation of Document Defect Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, pp. 99-107, 1996.

[17] G. Nagy, "Validation of OCR Data Sets," *Proc. Third Ann. Symp. Document Analysis and Information Retrieval,* pp. 127-135, Apr. 1994.

[18] I. Phillips, "User's Reference Manual," *CD-ROM, UW-III Document Image Database-III.*

[19] P. Vojta, *XDVI Software.* 1990.

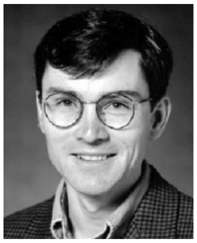


Tapas Kanungo received the MS and PhD degrees in electrical engineering from the University of Washington, Seattle, in 1990 and 1996, respectively. Currently, he serves as a codirector of the Language and Media Processing Lab at the University of Maryland, College Park, where he conducts research in the areas of document image analysis, OCR-based cross-language information retrieval, pattern recognition, and computer vision. From March 1996 to October 1997, he worked at Caere Corporation, Los Gatos, California on their OmniPage OCR product. During the summer of 1994, he worked at Bell Labs, Murray Hill, New Jersey and during the summer of 1993, he worked at the IBM Almaden Research Center, San Jose, California. Prior to that, from 1986 to 1988, he worked on speech coding and online handwriting analysis in the computer science group at Tata Institute for Fundamental Research, Bombay, India. He cochaired the 1999 IAPR Workshop on Multilingual OCR, was a coguest editor of the *International Journal of Document Analysis and Recognition*, special issue on performance evaluation, and has been program committee member several conferences. He is a member of IEEE.

Robert M. Haralick received the BA degree in mathematics in 1964, the BS degree in electrical engineering in 1966, and the MS degree in electrical engineering in 1967, all from the University of Kansas. In 1969, after receiving the PhD degree from the University of Kansas, he joined the faculty of the Electrical Engineering Department, where he was a professor from 1975 to 1978. In 1979, he joined the Electrical Engineering Department at Virginia Polytechnic Institute and State University where he was a professor and director of the Spatial Data Analysis Laboratory. From 1984 to 1986, he served as the vice president of research at Machine Vision International, Ann Arbor, Michigan. Dr. Haralick now occupies the Boeing Clairmont Egtvedt Professorship in the Department of Electrical Engineering at the University of Washington. He was elected a fellow of IEEE for his contributions in computer vision and image processing. He serves on the editorial boards of *Machine Vision and Applications* and *Real Time Imaging* and he is an associate editor for the *IEEE Transactions on Image Processing and Journal of Electronic Imaging*. His recent work is in shape analysis and extraction using the techniques of mathematical morphology and in robust pose estimation, techniques for making geometric inferences from perspective projection information, propagation of random perturbations through image analysis algorithms, and document analysis. He has developed the morphological sampling theorem that establishes a sound shape/size basis for the focus of attention mechanisms that can process image data in multiresolution mode, thereby making some of image feature extraction processes execute more efficiently.

Henry S. Baird is a principal scientist and manager of the document image analysis group at the Xereox Palo Alto Research Center. He has served as program cochair for the 1997 IAPR International Conference on Document Analysis and Recognition. He was general chair of the 1996 Symposium on Document Analysis and Information Retrieval and the 1990 IAPR Workshop on Syntactic and Structural Pattern Recognition. He is a member of the editorial board of the *International Journal on Document Analysis and Recognition* and an area editor for the *Computer Vision and Image Understanding* journal. His Princeton PhD thesis on algorithms for image matching won a 1984 ACM Distinguished Dissertation Award. In 1976, his Master's thesis gave the first complete description of the sweep-line algorithm, a fundamental technique in computational geometry. He has published three books and more than 50 technical articles. He is a senior member of the IAPR and also of the IEEE.

Werner Stuezle received the undergraduate degree from the Heidelberg University, the Master's (1973), and the PhD (1977) degrees, both in mathematics, from the Swiss Federal Institute of Technology (ETH) in Zurich. From 1978 to 1983, he served as an assistant professor in the Department of Statistics at Stanford University with a joint appointment in the Computation Research Group of the Stanford Linear Accelerator Center. In 1981, he was a visiting professor at the Department of Applied Mathematics and Center for Computational Research in Economics and Management Science at Massachusetts Institute of Technology. From 1983 to 1984, he was a research staff member at the IBM Zuerich Research Lab. Since 1984, he has been on the faculty in the Statistics Department at the University of Washington, Seattle. He has an adjunct appointment in the Department of Computer Science and Engineering and served as the chairman of the Statistics Department from 1994 to 1999.



David Madigan received the PhD degree in statistics from Trinity College, Dublin in 1990. Currently, he is vice president of Dialogue Mining at Soliloquy, Inc., a New York-based internet company. Previously, he was an associate professor of statistics at the University of Washington. He is a fellow of the American Statistical Association.