Deconvolution When Classifying Noisy Data Involving Transformations

Raymond Carroll
Department of Statistics, Texas A&M University, College Station, Texas 77843-3143, USA
carroll@stat.tamu.edu

Aurore Delaigle
Department of Mathematics and Statistics, University of Melbourne, VIC, 3010, Australia
A.Delaigle@ms.unimelb.edu.au

Peter Hall
Department of Mathematics and Statistics, University of Melbourne, VIC, 3010, Australia
halpstat@ms.unimelb.edu.au

Abstract. We consider the problem of classifying spatial data distorted by a linear transformation or convolution and contaminated by additive random noise. In this setting we show that classifier performance can be improved if we carefully invert the data before the classifier is applied. However, the inverse transformation is not constructed so as to recover the original signal, and in fact we show that taking the latter approach is generally inadvisable. We introduce a fully data-driven procedure based on crossvalidation, and use several classifiers to illustrate numerical properties of our approach. Theoretical arguments are given in support of our claims. Our procedure is applied to data generated by Lidar technology, where we improve on earlier approaches to classifying aerosols.

Keywords. Centroid classifier; Crossvalidation, Fourier transform, Inverse transform, Spatial data.

1 Introduction

We consider signal classification problems, where the observations are d-dimensional noisy spatial functions Y_{ij}, for $1 \le i \le n_j$, coming from population Π_j, where $j = 1$ or 2, and which can be modeled as $Y_{ij} = TX_{ij} + \delta_{ij}$, where T is a transformation of the function of interest X_{ij}, and δ_{ij} is a random error with zero mean and some correlation structure. Based on training data, the goal is to classify a new noisy data function Y, whose class is unknown, as coming from one of Π_1 and Π_2.

In many instances the function TX_{ij} is the result of a convolution of the function X_{ij} with a blurring source, that is, $TX_{ij} = \omega_T * X_{ij}$, where $*$ denotes the convolution operator (see Section 2.3) and ω_T is a point spread function. There, the function X_{ij} can be reconstructed in part by a (necessarily estimated) deconvolution operation. There is a large statistics literature on deconvolution for image data, and for data of similar type, dating from the

1980s. It includes contributions by Besag (1986), Donoho (1994), Dass and Nair (2003), Qiu (2005, 2007, 2008), and Mukherjee and Qiu (2011). Related research problems arise in spatial statistics, for example in the contexts of remote sensing (see e.g. Klein and Press, 1992; Cressie and Kornak, 2003; Crosilla et al., 2007) and statistical signal recovery (see e.g. Johnstone, 1990; Huang and Cressie, 2000; Shi and Cressie, 2007).

There is also a significant literature on blind deconvolution and estimation of point spread functions. This work includes contributions by Kundur and Hatzinakos (1998), Cannon (1976), Carasso (2001), Galatsanos et al. (2002), Figueiredo and Nowak (2003), Joshi and Chaudhuri (2005), Hall and Qiu (2007a,b), Qiu (2008), Huang and Qiu (2010) and Popescu and Hellicar (2010). However, the problems of deconvolution and point spread function estimation are very different from classification, to such an extent that, even if T were known, the methods suggested in this paper would still be recommended. It should also be noted that, since neither the function X nor the noise δ is observable, it is not possible to estimate the noise and, hence, to remove it effectively from the observed data Y. In particular, in the problem treated in this paper it is not possible to compute residuals.

In our classification context, it is at least intuitively plausible that if one could recover the function X_{ij}, then one would use that function as the basis for classification, rather than using the noisy convolved function Y_{ij}. This idea has been used in the classification of different types of aerosols using long range infrared light detection and ranging (Lidar) methods (Warren, et al., 2008), where deconvolution was used to obtain estimates of the true signal, and the resulting estimates were used as the basis for classification. Our work relates to whether a signal should be deconvolved or correlated errors should be deconvolved before classification, and we shall use Lidar data to illustrate our conclusions. We shall show that there exists a transformation of the noisy convolved function Y_{ij} that is appropriate for classification, but that it is not necessarily related to the transformation that would be used to recover the true signal.

The real-data classification problems that motivate this work all involve just $K = 2$ populations, and for this reason, and to simplify discussion, we shall confine attention to that case. However, out methodology and theoretical results extend readily to the general case $K \geq 2$, using the approach suggested by Friedman (1996).

The paper is organized as follows. We introduce our model and ideas in Section 2, and in Section 3 we establish theoretical properties of our procedure. In Section 4, using a variety of classifiers, we apply our approach to simulated data and to the Lidar data described above. Technical arguments are deferred to the **Supplementary Material**.

2 Methods

2.1 Model and classification problem

We observe spatial data functions $Y_{ij}(r)$, $r \in \mathcal{D}$, $1 \le i \le n_j$, $j = 1, 2$, generated by the model

$$Y_{ij}(r) = TX_{ij}(r) + \delta_{ij}(r), \tag{2.1}$$

where $\mathcal{D}$ denotes a d-dimensional spatial grid, or lattice, X_{ij} is the spatial function of interest, T is a linear transformation that blurs the signal, and δ_{ij}, representing noise, is a component of a correlated stochastic process with zero mean affecting the signal. In this model the data come from two populations, Π_1 or Π_2, and, for $j = 1, 2$, Y_{ij} denotes the i^{th} data function drawn from the j^{th} population Π_j, where $i = 1, ..., n_j$. To simplify notation we define scale in such a way that $\mathcal{D} \subset \mathbb{Z}^d$, where $\mathbb{Z}$ is the set of all integers.

The model at (2.1) is appropriate when the observations Y_{ij} are, for example, digitized images or Lidar signals. There, T typically represents the accumulated impact of issues such as generalized lens aberrations, atmospheric effects, motion blur, etc, and $\mathcal{D}$ is a two or three-dimensional grid.

Remark 1. It is important to realize that Y_{ij}, X_{ij} and δ_{ij} are functions defined on $\mathcal{D}$, and that T (and later the transforms R and Q that will be defined below) is not a function; it is a functional that maps the function X_{ij} to the function TX_{ij}.

Let Y be a new data value coming from Π_k, where $k = 1$ or 2 is unknown. Our goal is to construct a classifier $\mathcal{C}(\cdot) \equiv \mathcal{C}(\cdot \,|\, \{Y_{ij}\}_{j=1,2;i=1,...,n_j})$ from the data Y_{ij}, which assigns Y to $\Pi_{\hat{k}}$, where $\hat{k} = \mathcal{C}(Y \,|\, \{Y_{ij}\}_{j=1,2;i=1,...,n_j}) = 1$ or 2 is an estimator of k. In the applications we have in mind, where the data are images or Lidar signals, distinguishing between Π_1 and Π_2 is inherently a problem involving high dimensional data analysis. In practice, the number

of points r at which we observe data $Y_{ij}(r)$ can be in the thousands, whereas the training sample sizes, n_1 and n_2, are often only 20 or 30.

2.2 Deconvolution of the data through the noise transform

As we indicated in the introduction, when the functional T is invertible it is sometimes argued that, instead of applying standard classifiers to the data Y_{ij}, one should apply them to inverted data, where Y and each Y_{ij} are replaced by $T^{-1}Y$ and $T^{-1}Y_{ij}$, or rather by regularized versions of them, $\widehat{T}^{-1}Y$ and $\widehat{T}^{-1}Y_{ij}$. That is, classification should be based on $\mathcal{C}(\widehat{T}^{-1}Y \,|\, \{\widehat{T}^{-1}Y_{ij}\}_{j=1,2;i=1,\ldots,n_j})$ instead of $\mathcal{C}(Y \,|\, \{Y_{ij}\}_{j=1,2;i=1,\ldots,n_j})$. Transforming the data by T^{-1} is a good idea when the goal is to recover the function X_{ij}, since we have $T^{-1}Y_{ij} = X_{ij} + T^{-1}\delta_{ij}$, so that the transformed data are no longer distorted, and contain only additive noise $T^{-1}\delta_{ij}$ of zero mean. This is only approximately true when using $\widehat{T}^{-1}$, of course. See for example Cannon and Hunt (1981) and Hall (1990). However, we argue that when the goal is classification, inverting T is not necessarily a good idea, and a better strategy is to transform the data in such a way that classification performance is improved.

To explore the classification problem further, let $\epsilon_{ij} = TX_{ij} - T\mu_j + \delta_{ij}$ where the function μ_j is defined by $\mu_j = E_j(X_{ij})$ and E_j denotes expectation conditional on X_{ij} coming from population j. Then (2.1) can be written as

$$Y_{ij} = T\mu_j + \epsilon_{ij}\,, \tag{2.2}$$

where $E_j(\epsilon_{ij}) = E_j(\delta_{ij}) = 0$. If the processes $X_{ij} - \mu_j$ and δ_{ij} are also linear, in particular if ϵ_{ij} is stationary and Gaussian, as is often approximately the case in practice, then we can write $\epsilon_{ij} = R\,\xi_{ij}$ where R is another linear transformation and the process ξ_{ij} is white noise, i.e., the random variables $\xi_{ij}(r)$, for $r \in \mathbb{Z}^d$, are uncorrelated and have zero mean and common variance σ^2. In this notation the model at (2.2) can be expressed as

$$Y_{ij} = T\mu_j + R\,\xi_{ij}\,, \tag{2.3}$$

so that if R is invertible, (2.3) can be written equivalently as

$$R^{-1}Y_{ij} = R^{-1}T\mu_j + \xi_{ij}\,.$$

The absence of correlation of ξ_{ij}, and the constant variances, suggest that, for a variety of classifiers, performance can be improved by working with the data $R^{-1}Y$ rather than with Y itself. For example, this is the case when the error process ϵ_{ij} in (2.2) is stationary and Gaussian and we use the centroid classifier (see Section 3.1). Indeed, there the classifier based on such transformed data is Fisher's linear discriminant, albeit in a much higher dimensional setting than is usually contemplated, and so has optimality properties. In particular, this classifier is asymptotically equivalent to applying a likelihood-ratio test. More generally, we shall show in Section 3 that in non-Gaussian cases, the optimal transformation, in terms of asymptotic performance of the centroid classifier, is also R^{-1}.

These considerations suggest that, for such classifiers, far from it being a good idea to replace Y and Y_{ij} by their deconvolved forms $T^{-1}Y$ and $T^{-1}Y_{ij}$, we should replace them by $R^{-1}Y$ and $R^{-1}Y_{ij}$ and base classification on $\mathcal{C}(R^{-1}Y \,|\, \{R^{-1}Y_{ij}\}_{j=1,2;i=1,\ldots,n_j})$. For more general classifiers too, transforming the data prior to applying a classifier can often improve performance, but not when this transform is taken to be T^{-1}. In practice the optimal transform is unknown and is not necessarily equal to R^{-1} for each classifier, since the best transform may depend on the particular classifier in use. Likewise, the optimal transform is not necessarily always exactly linear. However, by inverting the Y_{ij}'s via a carefully chosen linear transform, which we shall denote by Q^{-1} in the next section, we can often improve classification performance significantly. We suggest such a practicable inversion technique in the next section, and we construct it from the data in such a way as to optimise classification peformance.

2.3 Transforming the data in practice

2.3.1 Modeling the transform

Since the best transform to apply to the data Y_{ij} prior to classification is generally not known, it needs to be estimated from the data. However, the sample size is usually too small for estimating this transform without imposing restrictions on it. Motivated by our discussion in the last paragraph of Section 2.2, we model the transform by the inverse Q^{-1} of a linear transform $Q = Q_\theta$, which depends on a low-dimensional vector of parameters

$\theta = (\theta_1, \ldots, \theta_q)$, as follows.

Let ω_{Q_θ} be a nonnegative weight function defined on $\mathbb{Z}^d$ and depending on θ. Moreover, let $*$ denote the discrete convolution operation, defined for any two absolutely square summable functions f and g by $f * g(r) = \sum_{s \in \mathbb{Z}^d} f(r-s)\, g(s)$. We take Q_θ to be the linear transform which maps a function ζ to a function $\chi_\theta = Q_\theta\, \zeta$ defined, for each $r \in \mathbb{Z}^d$, by

$$\chi_\theta(r) = \omega_{Q_\theta} * \zeta(r)\,. \tag{2.4}$$

In image analysis terminology, ω_{Q_θ} is called the spread function of the transform Q_θ. The choice of the parameters θ will be treated in Section 2.3.3.

An example of a simple model for ω_{Q_θ} is the two-parameter family $\omega_{p_0;\,\theta}$, where $\theta = (\rho, \ell)$ and $\omega_{p_0;\,\theta}$ is the ℓ-fold convolution of the probability mass function p_0, defined by

$$p_0(r) = \Big(\frac{1-\rho}{1+\rho}\Big)^d \rho^{|r|}\,, \quad r \in \mathbb{Z}^d\,, \tag{2.5}$$

where $|r| = \sum_{j=1}^d |r_j|$ and $|\rho| < 1$ (usually, $0 < \rho < 1$). This is the model we used in our numerical work in Section 4, but alternative models and more comments are given in appendix A.2 in the **Supplementary Material**.

2.3.2 Inverting Q

Since Q_θ is defined by a convolution, its inverse is more easily expressed in the Fourier domain. Let ζ be a function defined on $\mathbb{Z}^d$ such that $\sum_{r \in \mathbb{Z}^d} |\zeta(r)| < \infty$. The (discrete) Fourier transform $\phi_\zeta(t)$, for $t \in (-\pi, \pi)^d$, is defined by

$$\phi_\zeta(t) = \sum_{r \in \mathbb{Z}^d} \zeta(r)\, \exp\big(i\, r^{\mathrm{T}} t\big)\,, \tag{2.6}$$

where on this occasion $i = \sqrt{-1}$. Since the Fourier transform of a convolution between two functions is equal to the product of their Fourier transforms, we deduce from (2.4) that the Fourier transform of the function χ_θ is given by $\phi_{\chi_\theta} = \phi_\zeta\, \phi_{\omega_{Q_\theta}}$.

In this notation, when $\phi_{\omega_{Q_\theta}}(t) \neq 0$ we can write $\phi_\zeta(t) = \phi_{\chi_\theta}(t) / \phi_{\omega_{Q_\theta}}(t)$. If $|\phi_{\chi_\theta}| \,/\, |\phi_{\omega_{Q_\theta}}|$ is integrable then Q_θ is invertible, and the inverse transform Q_θ^{-1}, obtained by the Fourier inversion theorem, maps the function χ_θ into the function $Q_\theta^{-1}\chi_\theta$ defined by (2.8) below,

taking there $\mathcal{T} = (-\pi, \pi)^d$. If Q_θ is not invertible we can typically define a generalized inverse, Q_θ^{-1}, by truncating the integral used in Fourier inversion to a small enough set $\mathcal{T} \subset (-\pi, \pi)^d$, for example

$$\mathcal{T} = \{t : \|t\| \le \eta\} \quad \text{or} \quad \mathcal{T} = \{t : |t_j| \le \eta,\ 1 \le j \le d\}\,, \tag{2.7}$$

with $\eta \in (0, \pi)$. Thus, in either case, we can write

$$Q_\theta^{-1}\chi_\theta(r) = (2\pi)^{-d} \int_{\mathcal{T}} \exp\big(-i\, r^{\mathrm{T}} t\big) \left\{\phi_{\chi_\theta}(t) \,/ \phi_{\omega_{Q_\theta}}(t)\right\} dt\,. \tag{2.8}$$

Remark 2. To motivate the selections of $\mathcal{T}$ in (2.7), observe that $\phi_{\omega_{Q_\theta}}(0)$ equals the sum of the weights $\omega_{Q_\theta}(r)$ over $r \in \mathbb{Z}^d$, and the $\omega_{Q_\theta}(r)$'s would normally be chosen so that this sum was strictly positive, in fact equal to 1. Therefore $\phi_{\omega_{Q_\theta}}(0) \neq 0$, and by continuity, $\phi_{\omega_{Q_\theta}}(t) \neq 0$ for t in a sufficiently small neighborhood of the origin. Hence, choosing $\mathcal{T}$ as in the formulae in (2.7), for sufficiently small η, ensures that the integral at (2.8) is well defined if the function χ_θ is uniformly bounded.

For example, if we model Q_θ by taking $\omega_{Q_\theta} = \omega_{p_0;\theta}$, defined above (2.5), then $Q_\theta^{-1}\chi_\theta$ is particularly easy to calculate. As a matter of fact, by standard calculations we have

$$\phi_{\omega_{Q_\theta}}(t) = \phi_{\omega_{p_0;\theta}}(t) = \prod_{j=1}^{d} \left[1 + 2\rho\,(1-\rho)^{-2}\left\{1 - \cos(t_j)\right\}\right]^{-\ell}, \tag{2.9}$$

for each $t = (t_1, \ldots, t_d)^{\mathrm{T}} \in (-\pi, \pi)^d$, so that

$$Q_\theta^{-1}\chi_\theta(r) = (2\pi)^{-d} \sum_{s \in \mathbb{Z}^d} \chi_\theta(s) \int_{\mathcal{T}} \prod_{j=1}^{d} \Big(e^{i\,(s_j - r_j)t_j} \left[1 + 2\rho\,(1-\rho)^{-2}\left\{1 - \cos(t_j)\right\}\right]^{\ell} dt_j \Big)\,. \tag{2.10}$$

The integral in (2.10) is well defined if we take $\mathcal{T} = (-\pi, \pi)^d$, in which case it simplifies to

$$Q_\theta^{-1}\chi_\theta(r) = (2\pi)^{-d} \sum_{s \in \mathbb{Z}^d} \chi_\theta(s) \prod_{j=1}^{d} \int_{-\pi}^{\pi} e^{i\,(s_j - r_j)t_j} \left[1 + 2\rho\,(1-\rho)^{-2}\left\{1 - \cos(t_j)\right\}\right]^{\ell} dt_j\,. \tag{2.11}$$

A very attractive aspect of this choice of Q_θ is that we do not need smoothing parameters, such as η at (2.7), to regularize the integral. Further, it can be proved that each integral in (2.11) is equal to a constant depending only on $|s_j - r_j|$, ρ and ℓ, and which vanishes if $|s_j - r_j| > \ell$. In other words, $Q_\theta^{-1}\chi_\theta(r)$ is a linear combination of values of $\chi_\theta(s)$, for s in a neighborhood of r (more precisely, for s such that $\max_{j=1,\ldots,d} |s_j - r_j| \le \ell$).

2.3.3 Estimation of unknown parameters

Now that we have a practicable representation Q_θ^{-1} for the transform to apply to the data before classification, it remains to choose θ. Just as, a priori, it may seem natural to invert the data by T^{-1}, it may also seem natural to choose θ to give a good fit to the data. However, again our goal here is to classify, and thus θ should rather be chosen to optimize the performance of the classifier based on $\mathcal{C}_\theta(Y) \equiv \mathcal{C}(Q_\theta^{-1}Y \,|\, \{Q_\theta^{-1}Y_{ij}\}_{j=1,2;i=1,\ldots,n_j})$. We suggest choosing θ to minimize a crossvalidation estimator of error rate.

Specifically, write π_1 for the prior probability of Π_1, which is typically taken to equal $\frac{1}{2}$ if we have no a priori knowledge, or to $n_1/(n_1+n_2)$ if we believe that the proportion of observations from Π_1 in the training sample is representative of that in the population. Define

$$\widehat{e}(\theta) = \frac{\pi_1}{n_1}\sum_{i=1}^{n_1} I\{\mathcal{C}_{\theta;-i1}(Y_{i1}) = 2\} + \frac{1-\pi_1}{n_2}\sum_{i=1}^{n_2} I\{\mathcal{C}_{\theta;-i2}(Y_{i2}) = 1\}\,, \tag{2.12}$$

where $\mathcal{C}_{\theta;-ij}$ denotes the version of $\mathcal{C}_\theta$ constructed without using Y_{ij}, that is, $\mathcal{C}_{\theta;-ij}(Y_{ij}) = \mathcal{C}(Q_\theta^{-1}Y_{ij} \,|\, \{Q_\theta^{-1}Y_{k\ell}\}_{k=1,2;\ell=1,\ldots,n_k;(k,\ell)\neq(i,j)})$. Then $\widehat{e}(\theta)$ estimates the error rate,

$$e(\theta) = \pi_1\, P_1\{\mathcal{C}_\theta(Y) = 2\} + (1-\pi_1)\, P_2\{\mathcal{C}_\theta(Y) = 1\}\,, \tag{2.13}$$

where P_j denotes probability conditional on $Y \in \Pi_j$. We suggest choosing θ to minimize $\widehat{e}(\theta)$.

Remark 3. In cases where the set $\mathcal{T}$ cannot be taken equal to $(-\pi,\pi)^d$, the classifier can also depend on a small number of parameters defining $\mathcal{T}$, which, if they are unknown, can play the role of a smoothing parameter. See the examples at (2.7). In such cases, $\mathcal{C}_\theta$, $\widehat{e}(\theta)$ and $e(\theta)$ are replaced by $\mathcal{C}_{\theta,\mathcal{T}}$, $\widehat{e}(\theta,\mathcal{T})$ and $e(\theta,\mathcal{T})$, respectively, and θ and $\mathcal{T}$ are chosen to minimize $\widehat{e}(\theta,\mathcal{T})$.

3 Theory

3.1 Centroid classifier

There exist a variety of standard classifiers which give good performance for high dimensional data. Here we discuss detailed theoretical properties in the context of one of the most

popular and effective methods, the centroid-based technique; see for example James and Hastie (2001) and Shin (2008). If $\bar{Y}_j(r) = n_j^{-1} \sum_i Y_{ij}(r)$, the centroid method assigns a new value Y, coming from Π_1 or Π_2, to Π_1 (i.e., it puts $\mathcal{C}(Y) = 1$) if $\sum_{r \in \mathcal{D}} \left[\{Y(r) - \bar{Y}_2(r)\}^2 - \{Y(r) - \bar{Y}_1(r)\}^2 \right] > 0$, and to Π_2 (i.e., it puts $\mathcal{C}(Y) = 2$) otherwise. Other classifiers will be discussed in Section 4.4.

As already highlighed in section 2.2, if the errors are stationary then this classifier is optimised when applied to the data $R^{-1}Y_{ij}$. Using the representation Q_θ^{-1} for R^{-1}, an approximation to optimal classification consists of assigning a new observation Y to Π_1 (i.e., putting $\mathcal{C}_\theta(Y) = 1$) if and only if $S_\theta(Y) > 0$, where

$$S_\theta(Y) = \sum_{r \in \mathcal{D}} \Big\{ \big| Z_\theta(r) - \bar{Z}_{2;\theta}(r) \big|^2 - \big| Z_\theta(r) - \bar{Z}_{1;\theta}(r) \big|^2 \Big\}, \tag{3.1}$$

with $\bar{Z}_{j;\theta}(r) = n_j^{-1} \sum_i Z_{ij;\theta}(r)$ and where the functions Z_θ and $Z_{ij;\theta}$ are defined by $Z_\theta = Q_\theta^{-1} Y$ and $Z_{ij;\theta} = Q_\theta^{-1} Y_{ij}$.

In this notation the crossvalidation technique for choosing θ, described at equation (2.12) in section 2.3.3, can be written as

$$\widehat{e}(\theta) = \frac{\pi_1}{n_1} \sum_{i=1}^{n_1} I\{S_{\theta;-i1}(Y_{i1}) \le 0\} + \frac{1 - \pi_1}{n_2} \sum_{i=1}^{n_2} I\{S_{\theta;-i2}(Y_{i2}) > 0\}\,, \tag{3.2}$$

where $S_{\theta;-ij}$ denotes the version of S_θ at (3.1) calculated with $\bar{Z}_j$ replaced by $\bar{Z}_j^{(-i)} = (n_j - 1)^{-1} \sum_{k \ne i} Z_{kj}$. Likewise, the error rate $e(\theta)$ at (2.13) can be written as

$$e(\theta) = \pi_1 \, P_1\{S_\theta(Y) \le 0\} + (1 - \pi_1) \, P_2\{S_\theta(Y) > 0\}\,. \tag{3.3}$$

3.2 Main assumptions

To simplify notation, throughout Section 3 we define scale in such a way that $\mathcal{D}$, in d-variate Euclidean space, has edge width 1, for example:

$$\mathcal{D} = \Big\{ r = (r_1, \ldots, r_d)^{\mathrm{T}} : r_1, \ldots, r_d \in \mathbb{Z},\ -n \le r_1, \ldots, r_d \le n \Big\}\,, \tag{3.4}$$

where $n \ge 1$. In this setting, $\#\mathcal{D} \asymp n^d$ and the training sample sizes, n_1 and n_2, are interpreted as functions of n. Let $\mathcal{Y}$ denote the pair of training samples $(\mathcal{Y}_1, \mathcal{Y}_2)$ with

$\mathcal{Y}_j = \{Y_{ij},\, 1 \le i \le n_j\}$, $Y_{ij} = (Y_{ij}(r) : r \in \mathcal{D})$. The error rate of our classifier, computed from the training data set $\mathcal{Y}$, is denoted by $e(\theta)$ and defined at (3.3). In this section we give asymptotic formulae for $e(\theta)$ and $\widehat{e}(\theta)$, taking $\mathcal{T}$ to be a general subset of $(-\pi, \pi)^d$. For example, $\mathcal{T}$ might be equal to $(\pi, \pi)^d$, or to one of the regions defined at (2.7). Theory in cases where crossvalidation is used to determine $\mathcal{T}$, as well as θ (see Remark 3 in Section 2.3.3) can be developed at the expense of longer arguments; in the present section we use crossvalidation to optimize over θ but not $\mathcal{T}$, which corresponds to our practical implementation of the method; see Section 4.

We develop our theory under three main model assumptions. First, we assume that R maps a function ζ, defined on $\mathbb{Z}^d$, into a function $R\,\zeta$, defined by

$$R\,\zeta(r) = \omega_R * \zeta(r)\,. \tag{3.5}$$

Second, we assume that

$$R^{-1}\,T\mu_j, \quad Q_\theta^{-1}\,T\mu_j \ \text{ and } \xi \text{ are supported on } \mathcal{D}\,, \text{ for } j = 1, 2. \tag{3.6}$$

We impose this condition only to avoid long arguments for dealing with potential edge effects. Our conclusions remain valid without it, but the proofs become considerably longer. Finally, we assume that $T\mu_1 - T\mu_2 = T(\mu_1 - \mu_2)$ is smoother than ω_R. More precisely, we assume that $T(\mu_1 - \mu_2) = \alpha\, K * \omega_R$, where α is a constant and K is a function supported on $\mathcal{D}$. This assumption ensures that the inverse of the mean of the differences of the observed signals, $R^{-1}T(\mu_1 - \mu_2)$, remains bounded. It is imposed only to make our technical arguments simpler and explicit. If it is not satisfied, then, generally speaking, the classification problem becomes simpler, in that the difference between the means of the inverted signals is even larger and therefore easier to detect.

We allow the distance between the two transformed means, $T\mu_1$ and $T\mu_2$, to vary with n, by letting α above depend on n. In particular, we assume that $T(\mu_1 - \mu_2) = \alpha_n\, K * \omega_R$, where α_n is a sequence of positive real numbers bounded above zero. The most important case is that where α_n (and hence the distance) decreases with increasing n, since that enables our theoretical arguments to address particularly challenging cases. We also permit the noise variance, $\sigma_n^2 = \text{var}\{\xi_{ij}(r)\}$, to depend on n. We shall see that the relative sizes of n, α_n and

σ_n interact together to determine the performance of our classifier. Although this interaction is quite complex, to a large extent it can be represented in terms of the quantity

$$u_n(\theta) = (\alpha_n/\sigma_n) \int_{\mathcal{T}} |\phi_K|^2 \, |\phi_{\omega_R}|^2 \, |\phi_{\omega_{Q_\theta}}|^{-2} \Big/ \Big\{ (2\pi)^d \int_{\mathcal{T}} |\phi_K|^2 \, |\phi_{\omega_R}|^4 \, |\phi_{\omega_{Q_\theta}}|^{-4} \Big\}^{1/2}, \tag{3.7}$$

where $\phi_{\omega_R}(t) = \sum_{r \in \mathbb{Z}^d} \omega_R(r) \exp\left(i\, r^{\mathrm{T}} t\right)$ is the Fourier transform of ω_R, and $\phi_K(t) = \sum_{r \in \mathcal{D}} K(r) \exp\left(i\, r^{\mathrm{T}} t\right)$ is the Fourier transform of K; here we used the fact that K is supported on $\mathcal{D}$.

In order to derive our theoretical results we also need regularity conditions. These are more technical, and we shall describe them in detail in Appendix B.1 in the **Supplementary Material**; see (B.2) to (B.6).

3.3 Asymptotic formula for error rate

The next theorem describes properties of $e(\theta)$ as n diverges. Let Φ denote the standard normal distribution function and write Θ for a compact set of parameters from which θ is chosen.

Theorem 1. *Assume that the data are generated by the model at* (2.3)*, where R is of the form at* (3.5) *and T is a linear transformation, and that* (B.2)–(B.6) *hold. Then,*

$$\sup_{\theta \in \Theta} \left| e(\theta) - \Phi\{-u_n(\theta)\} \right| \to 0, \tag{3.8}$$

where the convergence is in probability.

To elucidate the implications of Theorem 1, observe first that the asymptotic error rate, $\Phi(-u_n)$, in (3.8) is a monotone decreasing function of u_n. It therefore follows from formula (3.7) for $u_n(\theta)$ that the error rate decreases as either the distance, represented by α_n, between population means increases, or the error variance, σ_n^2, decreases. Moreover, Hölder's inequality implies that $\Phi\{-u_n(\theta)\}$, interpreted as a functional of $\phi_{\omega_{Q_\theta}}$, is minimized when $\phi_{\omega_{Q_\theta}} = \phi_{\omega_R}$, i.e. when the transformation Q_θ is identical to the actual transformation R.

3.4 Consistency of crossvalidation estimator of error rate

Recall the definition of $\widehat{e}(\theta)$, the crossvalidation estimator of error rate, at (3.2). Theorem 2 below shows that $\widehat{e}(\theta)$ shares the same asymptotic property, (3.8), as the actual error rate

$e(\theta)$, and therefore is consistent for $e(\theta)$, uniformly in θ.

Theorem 2. *Assume the conditions of Theorem 1. Then,*

$$\sup_{\theta\in\Theta} \left|\widehat{e}(\theta) - \Phi\{-u_n(\theta)\}\right| \to 0\,, \tag{3.9}$$

where the convergence is in probability.

Similarly it can be proved that if $\theta = \widehat{\theta}$ is chosen to minimize $\widehat{e}(\theta)$, and used when constructing the classifier, then, under mild additional assumptions, the classifier's actual error rate will equal $\min_{\theta\in\Theta} \Phi\{-u_n(\theta)\} + o(1)$ as $n \to \infty$.

4 Numerical work

4.1 Goals of simulations

We performed simulation studies to illustrate the following properties:

(1) Transforming the data by T^{-1} prior to applying a classifier generally does not improve classification performance;

(2) Transforming the data using a crossvalidation-based transform $\widehat{Q}_\theta^{-1}$ generally improves classification performance, even if Q_θ^{-1} is only a rough approximation to the best transform to apply;

(3) The more the errors ϵ_{ij} are correlated, the larger is the improvement at (2), especially if the error variance σ^2 is large compared to $T\mu_j$;

(4) The performance of classifiers, applied to data transformed by $\widehat{Q}_\theta^{-1}$, improves as the training sample size and/or the fineness of the grid $\mathcal{D}$ increases.

4.2 Simulation setup

4.2.1 Generation of training samples

We generated training samples $\{Y_{11}, \ldots, Y_{1n_1}\}$ and $\{Y_{21}, \ldots, Y_{2n_2}\}$, of sizes $n_1 = n_2 = 10$ or $n_1 = n_2 = 25$, according to the model

$$Y_{ij}(r) = T\mu_j(r) + R\,\xi_{ij}(r), \tag{4.1}$$

for different curves μ_j, $j = 1, 2$, and transformations R and T, and with $r \in \mathcal{D} \subset \mathbb{R}$ or $r = (r_1, r_2) \in \mathcal{D} \subset \mathbb{R}^2$.

We considered four pairs of mean curves μ_j, for $j = 1, 2$ (two univariate and two bivariate), each with several features such as asymmetric peaks and valleys, or sinusoidal components:

(a) $\mu_j(r) = |2r - a_j|^{4/5} \exp\{-5 \cdot 10^{-4}(4r^2 - b_j)\}$, where $a_1 = 5, b_1 = 100, a_2 = 4, b_2 = 80$;

(b) $\mu_j(r) = 9/16 \cdot c_j^{-2}(2r - 50)^2 / \{1.2 + \cos(r)\}^2$, where $c_1 = 200, c_2 = 190$;

(c) $\mu_j(r_1, r_2) = |3r_2 - a_j|^{2/5} \exp\{-45 \cdot 10^{-4}(r_1^2 + 2r_1 r_2 + r_2^2 - b_j/9)\}$, with a_j and b_j as in (a);

(d) $\mu_j(r_1, r_2) = 0.1\,|4 + 3r_2/50|^{1/5} \cdot \exp\{-(3r_1 + 20)/d_j\}/\{1.2 + \cos(1.5r_1)\} \cdot 1_{[-20/3,\infty)}(r_1)$,
where $d_1 = 40, d_2 = 50$ and $1_{[-20/3,\infty)}(r_1) = 1$ if $r_1 \in [-20/3, \infty)$ and 0 otherwise.

In the previous sections, the method was discussed for a grid that had edge width 1. More generally, in our simulations we also considered examples where the grid has edge width $k_{\mathcal{I}}$. In that case, the various transformations have to be rescaled by a factor $k_{\mathcal{I}}$. More precisely, if a transform F has the form $F\,\xi(r) = \sum_{s \in \mathbb{Z}^d} \omega_F(s)\,\xi(r-s)$ on a grid of edge width 1, on a grid of edge width $k_{\mathcal{I}}$ it becomes $F\,\xi(r) = k_{\mathcal{I}}^d \sum_{s \in \mathbb{Z}^d_{k_{\mathcal{I}}}} \omega_F(s)\,\xi(r - s)$, where $\mathbb{Z}_{k_{\mathcal{I}}} = \{s/k_{\mathcal{I}}, s \in \mathbb{Z}\}$. Reflecting this discussion, we took $T\mu_j(r) = k_{\mathcal{I}}^d \sum_{s \in \mathbb{Z}^d_{k_{\mathcal{I}}}} \omega_{p_0;\,\theta_T}(s)\,\mu_j(r - s)$ and

$$R\,\xi(r) = k_{\mathcal{I}}^d \sum_{s \in \mathbb{Z}^d_{k_{\mathcal{I}}}} \omega_{p_0;\,\theta_R}(s)\,\xi(r - s), \tag{4.2}$$

where $\omega_{p_0;\,\theta}$ is the function defined above (2.5), with $\theta = \theta_T = (\ell_T, \rho_T)$ or $\theta = \theta_R = (\ell_R, \rho_R)$. In our bivariate models (c) and (d), we also considered

$$R\,\xi(r_1, r_2) = k_{\mathcal{I}}^2 \sum_{s \in \mathbb{Z}^2_{k_{\mathcal{I}}},\ |s_j + r_j| \le \theta_M / k_{\mathcal{I}}} \omega_M(|s_1 + r_1|)\,\omega_M(|s_2 + r_2|)\,\xi(r_1 - s_1, r_2 - s_2), \tag{4.3}$$

where, for $u \in \mathbb{Z}^+$, $\omega_M(u) = (\theta_M + 1 - u) / \sum_{u \le \theta_M}(\theta_M + 1 - u)$, with θ_M a positive integer.

In each case we considered several different values of θ_R in (4.2), or θ_M in (4.3), and we took the $\xi_{ij}(r)$ to be independent normal $\mathrm{N}(0, \sigma^2)$. Each combination of σ and θ_R or θ_M was chosen such that good classification was possible for at least one of the versions of the centroid classifier described below; see Tables 1 to 3 in Section A.3.1, in the **Supplementary Material**, for all the combinations we considered in practice, and for a measure of signal to

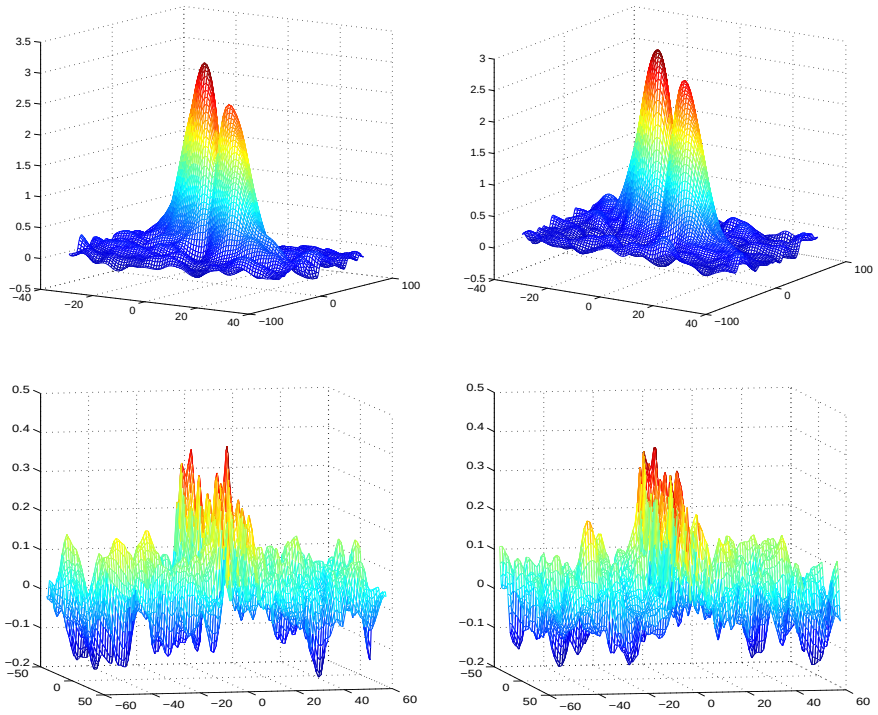

Figure 1: Plotted are Y_{11} (left) and Y_{12} (right), in Models (c) (row 1) and (d) (row 2) with $\theta_R = (0.5, 3)$.

noise ratio in each case. Finally, we took the parameter θ_T of the transform T, and the grid $\mathcal{D}$ where the data are observed, as follows:

- Model (a): $\theta_T = (0.5, 3)$ and $\mathcal{D} = \{-80, -80 + k_{\mathcal{I}}, \ldots, 80 - k_{\mathcal{I}}, 80\}$;
- Model (b), $\theta_T = (0.5, 2)$ and $\mathcal{D} = \{-80, -80 + k_{\mathcal{I}}, \ldots, 80 - k_{\mathcal{I}}, 80\}$;
- Models (c) and (d): $\theta_T = (0.25, 2)$ and $\mathcal{D} = \{-60, -60 + k_{\mathcal{I}}, \ldots, 60 - k_{\mathcal{I}}, 60\} \times \{-40, -40 + k_{\mathcal{I}}, \ldots, 40 - k_{\mathcal{I}}, 40\}$.

In each case, $k_{\mathcal{I}} = 2$ when $n_1 = n_2 = 10$ and $k_{\mathcal{I}} = 1$ or 2 when $n_1 = n_2 = 25$. In particular, when n_1 and n_2 were increased we let the grid $\mathcal{D}$ become finer by decreasing $k_{\mathcal{I}}$ from 2 to 1, so as to illustrate point (4) in Section 4.1. We also ran simulations in the unbalanced case, where $n_1 = 10$ and $n_2 = 25$, and obtained results similar to those that we shall discuss below. See Figures 4 and 5 of Section A.3.4 in the **Supplementary Material**.

For illustration, Figure 1 shows Y_{11} and Y_{12} in Models (c) and (d), with $\theta_R = (0.5, 3)$. Comparing with Figure 9 in Section 4.5, we can see that example (d) looks similar to our empirical example discussed in Section 4.5.

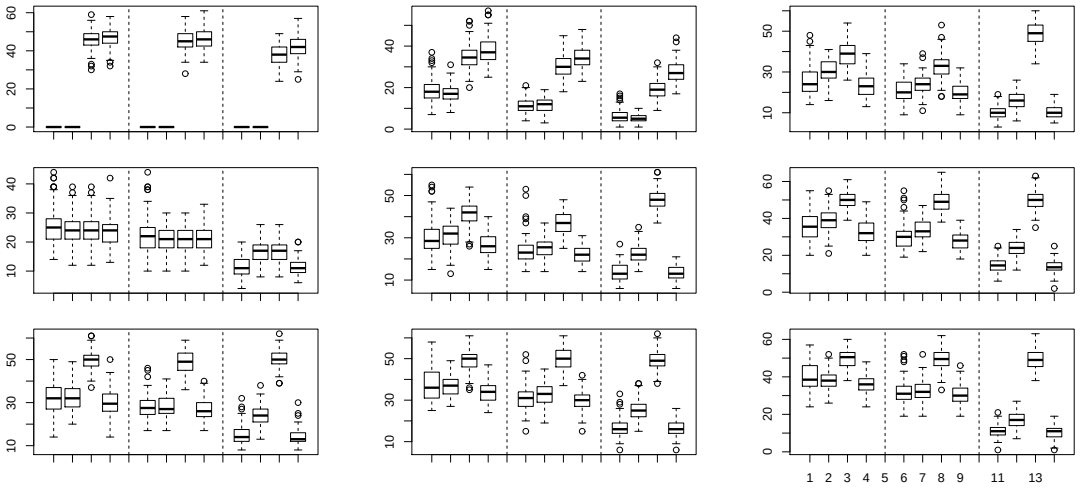

Figure 2: Boxplots of percentage of misclassified observations calculated from 100 simulated samples from model (a) when $\theta_R = (\rho_R, \ell_R)$, with $\rho_R = 0.75$, 0.5 and 0.25 in rows 1, 2 and 3, respectively, and $\ell_R = 3$, 2 and 1 in columns 1, 2 and 3, respectively. In each group of 12 boxes, the first four are for $n_1 = n_2 = 10$ and $k_{\mathcal{I}} = 2$, the next four are for $n_1 = n_2 = 25$ and $k_{\mathcal{I}} = 2$, and the last four are for $n_1 = n_2 = 25$ and $k_{\mathcal{I}} = 1$. In each group of four boxes, the data are transformed by $\widehat{Q}^{-1}$ (first box), R^{-1} (second box), T^{-1} (third box), or untransformed (fourth box).

4.2.2 Model for Q_θ, generation of test samples and estimation of error rate

No matter what model we used for R, we systematically modeled Q_θ by

$$Q_\theta\, \xi(r) = k_{\mathcal{I}}^d \sum_{s \in \mathbb{Z}_{k_{\mathcal{I}}}^d} \omega_{p_0;\,\theta}(s)\, \xi(r-s), \tag{4.4}$$

with $\theta = (\ell_Q, \rho_Q)$. This model is flexible, and, as discussed in details in Section 2.3, it has attractive practical properties such as the fact that we do not need any smoothing parameters to define $\mathcal{T}$ in (2.8), which can be taken equal to $\mathcal{T} = (-\pi, \pi)^d$.

To test our classifier constructed from the training observations Y_{ij}, we generated test samples of $N = 100$ new data curves $Y_1^{\text{New}}, \ldots, Y_{100}^{\text{New}}$, of which half came from Π_1 and the other half from Π_2, using each time the same model as the one used to generate the Y_{ij}'s. We applied several classifiers to three versions of the Y_i^{New}'s: the untransformed noisy data Y_i^{New}, the data $T^{-1}Y_i^{\text{New}}$, and the data $\widehat{Q}^{-1}Y_i^{\text{New}}$, where $\widehat{Q}$ denotes $Q_{\widehat{\theta}_{CV}}$, with Q_θ as at (4.4), and with $\theta = \widehat{\theta}_{CV}$ chosen to minimize the crossvalidation estimator of the classification error rate as in Section 2.3.3, where we took $\pi_1 = n_1/(n_1+n_2)$. When R was of the form at (4.2) we also applied the classifiers to the data $R^{-1}Y_i^{\text{New}}$.

As indicated above, we chose $\theta = (\rho, \ell)$ to minimize the crossvalidation estimator of classification error rate, where we performed the minimization over a bivariate grid of values in the

range $0 \leq \rho \leq 0.95$ and $1 \leq \ell \leq 5$. Here, $\rho = 0$ denotes the identity transform, and when $\rho = 0$ we do not transform the data. Observe that, in our simulations and examples, the sizes of the training data sets are small, and there is little computational cost. In larger data sets one would use k-fold crossvalidation, i.e., the training data would consist of a randomly selected $(1-k^{-1}) \times 100\%$ of the data, and the test data the remaining $(100/k)\%$, with this procedure repeated many times to calculate an overall error rate. Wikipedia has a good description of this approach (`http://en.wikipedia.org/wiki/Cross-validation_(statistics)`), and references McLachlan et al. (2004).

In practice, the transform T is often unknown and is not necessarily invertible. In such cases, instead of using T^{-1} one has to use a regularized estimator $\widehat{T}^{-1}$ constructed from the data (see our real data illustration). Here, for simplification we take T both known and invertible. While this may seem to be unfavorable to our approach, it actually does not matter since our point is to show that T^{-1} has essentially no role to play in our classification problem, and whether T^{-1} is known or estimated does not change our conclusions.

In each model we generated $B = 100$ training samples, and for each training sample we generated a test sample of $N = 100$ new data curves as described above, which we classified in one of the two populations using each of the methods described in the previous paragraph. For each training sample we calculated the percentage of the new curves that were misclassified by each method. We obtained $B = 100$ misclassification percentages for each method, and the boxplots shown below were computed from these 100 percentages.

4.3 Simulation results for centroid classifier

4.3.1 Data coming from the model in (4.1)

We start by reporting results obtained when applying the centroid classifier described in section 3.1 to data generated from the model in (4.1). In cases where $\widehat{e}$ achieved its minimum at several values θ, we broke the ties according to the rule described in Section A.1 in the **Supplementary Material**. The boxplots corresponding to each of the four methods described above, for R of the form at (4.2), are shown in Figures 2, 3 and 4. We present the results for various values of $\theta_R = (\rho_R, \ell_R)$, for $n_1 = n_2 = 10$ and $k_{\mathcal{I}} = 2$, $n_1 = n_2 = 25$ and

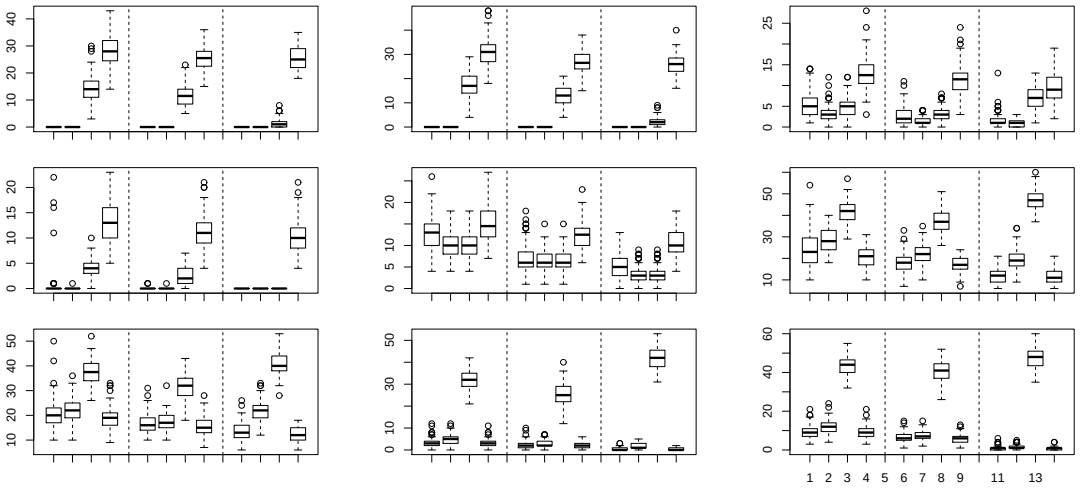

Figure 3: Boxplots of percentage of misclassified observations calculated from 100 simulated samples from model (b) when $\theta_R = (\rho_R, \ell_R)$, with $\rho_R = 0.75$, 0.5 and 0.25 in rows 1, 2 and 3, respectively, and $\ell_R = 3$, 2 and 1 in columns 1, 2 and 3, respectively. In each group of 12 boxes, the first four are for $n_1 = n_2 = 10$ and $k_{\mathcal{I}} = 2$, the next four are for $n_1 = n_2 = 25$ and $k_{\mathcal{I}} = 2$, and the last four are for $n_1 = n_2 = 25$ and $k_{\mathcal{I}} = 1$. In each group of four boxes, the data are transformed by $\widehat{Q}^{-1}$ (first box), R^{-1} (second box), T^{-1} (third box), or untransformed (fourth box).

$k_{\mathcal{I}} = 2$, and for $n_1 = n_2 = 25$ and $k_{\mathcal{I}} = 1$, where $k_{\mathcal{I}}$ is the distance between two adjacent univariate components of the grid $\mathcal{D}$. Our finite sample results support our asymptotic theory, which implies that as n_1 and n_2 increase (i.e. as training sample size increases) and $k_{\mathcal{I}}$ decreases (i.e. as the grid $\mathcal{D}$ becomes finer), the best results should be obtained by the centroid classifier applied to the data inverted by R^{-1}, of which $Q^{-1}_{\widehat{\theta}_{CV}}$ is a consistent estimator.

Overall, our results indicate that, in finite samples, it is the latter crossvalidation approach that is the most competitive. This is because this method has the ability to optimize performance based on the particular sample at hand. Unsurprisingly, transforming the data through R^{-1} and $Q^{-1}_{\widehat{\theta}_{CV}}$ brings the most significant improvements when ρ_R and ℓ_R are the largest, since it is in those cases that the correlation among the ϵ_{ij}'s is the largest. For smaller values of ρ_R and ℓ_R (e.g. $\rho_R = 0.25$ or $\ell_R = 1$), the correlation among the ϵ_{ij}'s is relatively small, and as a result, in finite samples the centroid method applied to the untransformed data Y^{New}_{ij} is often the most competitive approach, although even in those cases, the crossvalidation approach remains highly competitive. Of course in practice we do not know the transformation R, and our results indicate that crossvalidation-based inversion is the method of choice.

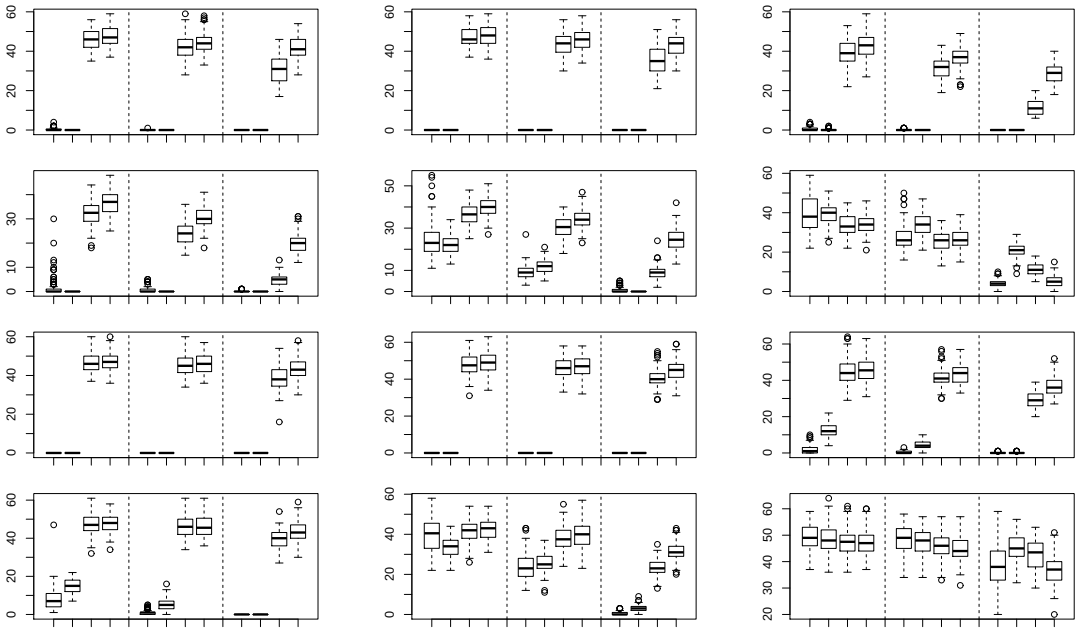

Figure 4: Boxplots of percentage of misclassified observations calculated from 100 simulated samples from models (c) (rows 1 and 2) and (d) (rows 3 and 4) when $\theta_R = (\rho_R, \ell_R)$, with $\rho_R = 0.85$ and 0.5 in rows 1,3 and 2,4 respectively, and $\ell_R = 3$, 2 and 1 in columns 1, 2 and 3, respectively. In each group of 12 boxes, the first four are for $n_1 = n_2 = 10$ and $k_{\mathcal{I}} = 2$, the next four are for $n_1 = n_2 = 25$ and $k_{\mathcal{I}} = 2$, and the last four are for $n_1 = n_2 = 25$ and $k_{\mathcal{I}} = 1$. In each group of four boxes, the data are transformed by $\widehat{Q}^{-1}$ (first box), R^{-1} (second box), T^{-1} (third box), or untransformed (fourth box).

4.3.2 Robustness against misspecification of R

Next we illustrate the robustness of the inversion procedure by reporting the results obtained when applying the centroid classifier to the data $Q^{-1}_{\widehat{\theta}_{CV}} Y_i^{\text{New}}$, with Q as at (4.4), when the true transform R was of another form, specifically the one at (4.3), where we took $\theta_M = 10$, 20 or 30. We compare this approach with the centroid-based classifier based on the data $T^{-1}Y_i^{\text{New}}$ and with the one based on the data Y_i^{New}. We show boxplots of the percentage of misclassified data curves in Figure 5, for each of the three methods and for $n_1 = n_2 = 10$ and $k_{\mathcal{I}} = 2$, $n_1 = n_2 = 25$ and $k_{\mathcal{I}} = 2$, and for $n_1 = n_2 = 25$ and $k_{\mathcal{I}} = 1$. Our results indicate that even if Q_θ at (4.4) is not the exact noise transformation, inverting the data through $Q^{-1}_{\widehat{\theta}_{CV}}$ can considerably improve on the centroid classifier based on either $T^{-1}Y_i^{\text{New}}$ or on the untransformed data Y_i^{New}.

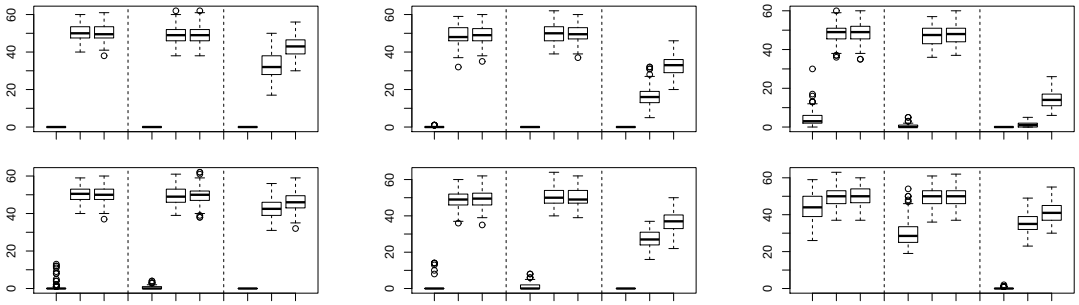

Figure 5: Boxplots of percentage of misclassified observations calculated from 100 simulated samples from models (c) (row 1) and (d) (rows 2) when R is of the form at (4.3), with $\theta_M = 30$, 20 and 10 in columns 1, 2 and 3, respectively. In each group of 9 boxes, the first three are for $n_1 = n_2 = 10$ and $k_{\mathcal{I}} = 2$, the next three are for $n_1 = n_2 = 25$ and $k_{\mathcal{I}} = 2$, and the last three are for $n_1 = n_2 = 25$ and $k_{\mathcal{I}} = 1$. In each group of three boxes, the data are transformed by $\widehat{Q}^{-1}$ (first box), T^{-1} (second box), or untransformed (third box).

4.3.3 Robustness against the stationarity assumption

In practice, the model at (4.1) is often an approximation to the model that generated the data. In this section, to investigate the effect of non stationarity of the errors on our procedure, we report results of simulations where the data Y_{ij} were generated from the model

$$Y_{ij}(r) = T\mu_j(r) + R_r\,\xi_{ij}(r), \tag{4.5}$$

with the fixed transform R replaced by a transform R_r depending on r.

In the univariate case, instead of R in (4.2), we used

$$R_r\,\xi(r) = k_{\mathcal{I}}^d \sum_{s\in\mathbb{Z}^d_{k_{\mathcal{I}}}} \omega_{p_0;\,\theta_r}(s)\,\xi(r-s), \tag{4.6}$$

with $\theta_r = (\rho_r, \ell)$, where $\rho_r = \rho + 0.1\cos(r/\alpha)$ (we considered two cases: $\alpha = 2$ and $\alpha = 10$) and ρ and ℓ as in the previous section. In the bivariate case, instead of using the transform R at (4.3) with constant θ_M, we used the transform

$$R_r\,\xi(r_1, r_2) = k_{\mathcal{I}}^2 \sum_{s\in\mathbb{Z}^2_{k_{\mathcal{I}}},\ |s_j+r_j|\le\theta_{M,r_j}/k_{\mathcal{I}}} \omega_{M,r_1}(|s_1+r_1|)\,\omega_{M,r_2}(|s_2+r_2|)\,\xi(r_1-s_1, r_2-s_2), \tag{4.7}$$

where, for $u \in \mathbb{Z}^+$ and $j = 1, 2$, $\omega_{M,r_j}(u) = (\theta_{M,r_j} + 1 - u)/\sum_{u\le\theta_{M,r_j}}(\theta_{M,r_j} + 1 - u)$, with $\theta_{M,r_j} = \theta_M + 2\cdot[\alpha\cos(r_j/2)]$ (we considered two cases: $\alpha = 2$ and $\alpha = 4$), θ_M as in the previsous section, and, for any real number x, we use $[x]$ to denote the integer closest to x.

Although here the errors $R_r\,\xi_{ij}(r)$ were non stationary, we inverted the data in the same way as before, using the transform $Q^{-1}_{\widehat{\theta}_{CV}}$. Figures 6 and 7 show boxplots of the percentage

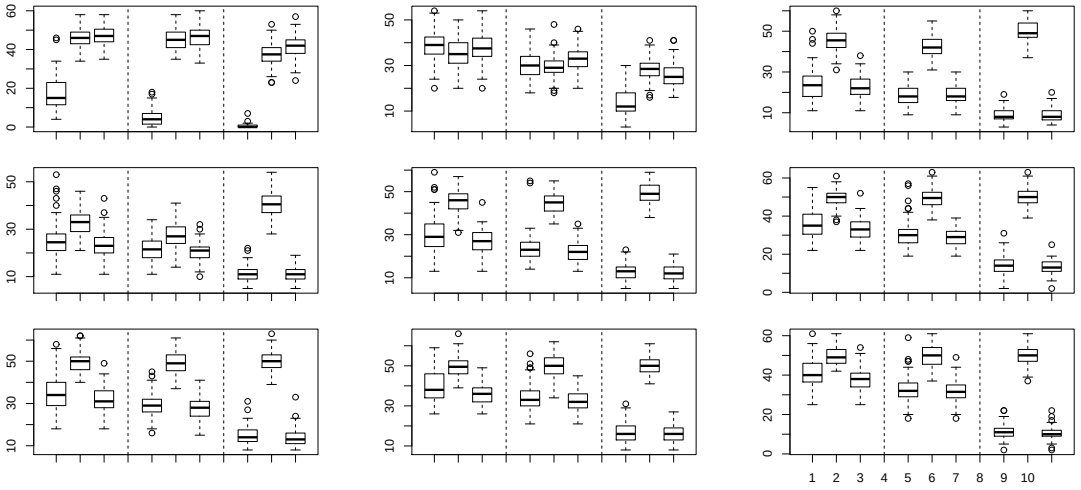

Figure 6: Boxplots of percentage of misclassified observations calculated from 100 simulated samples from model (a) when $\theta_R = (\rho_{R,r}, \ell_R)$, with $\rho_{R,r} = \rho_R + 0.1\cos(r/2)$, $\rho_R = 0.75$, 0.5 and 0.25 in rows 1, 2 and 3, respectively, and $\ell_R = 3$, 2 and 1 in columns 1, 2 and 3, respectively. In each group of 9 boxes, the first three are for $n_1 = n_2 = 10$ and $k_{\mathcal{I}} = 2$, the next three are for $n_1 = n_2 = 25$ and $k_{\mathcal{I}} = 2$, and the last three are for $n_1 = n_2 = 25$ and $k_{\mathcal{I}} = 1$. In each group of three boxes, the data are transformed by $\widehat{Q}^{-1}$ (first box), T^{-1} (second box), or untransformed (third box).

of missclassified curves for the centroid classifier constructed from the data $Q^{-1}_{\widehat{\theta}_{CV}} Y_{ij}$, Y_{ij} and $T^{-1}Y_{ij}$, where Y_{ij} was generated as in (4.5) with μ_j from model (a) and model (b), respectively, R_r as in (4.6) and $\alpha = 2$. For the case $\alpha = 10$, see Figures 1 and 2 in the **Supplementary Material**. Figure 8 shows similar results for the bivariate examples (c) and (d), when the data were generated according to (4.5) with R_r as in (4.7) and $\alpha = 2$. See Figure 3 in the **Supplementary Material** for the case $\alpha = 4$. These results indicate that our inversion method can improve classification performance significantly even when the errors are not exactly stationary; it usually does not degrade performance more than a little.

4.4 Other classifiers

Although it is beyond the scope of this paper to develop theory for all types of classifiers, and derive the theoretically optimal transform for each of them, we argue that our conclusions extend to other classifiers. To illustrate this, we also implemented two other classifiers often employed in high dimensional and functional data problems, which we applied to the four versions of the data: Y_{ij}, $T^{-1}Y_{ij}$, $R^{-1}Y_{ij}$ and $Q^{-1}_{\widehat{\theta}_{\rm CV}} Y_{ij}$, with $\widehat{\theta}_{\rm CV}$ chosen to minimize the crossvalidation estimate of classification error. Namely, we used the Support Vector Machine

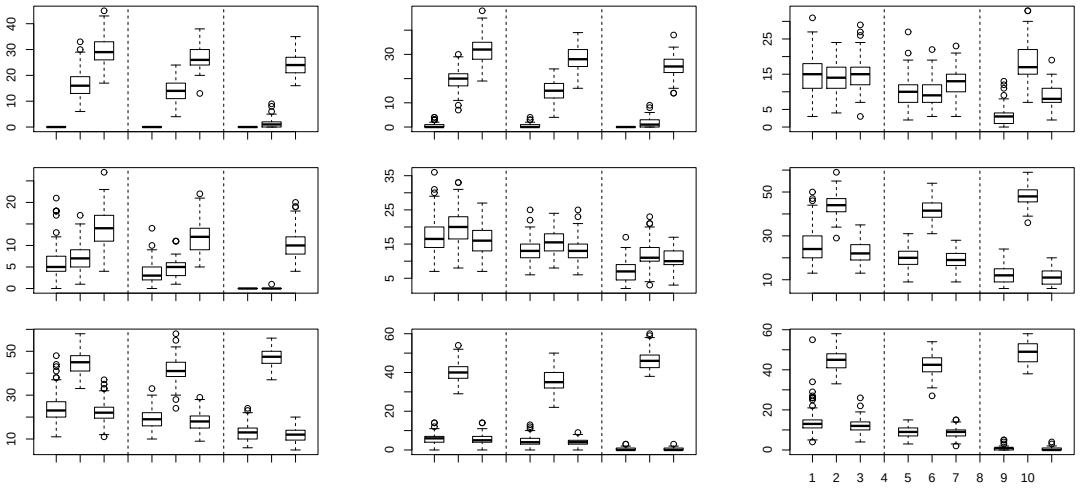

Figure 7: Boxplots of percentage of misclassified observations calculated from 100 simulated samples from model (b) when $\theta_R = (\rho_{R,r}, \ell_R)$, with $\rho_{R,r} = \rho_R + 0.1\cos(r/2)$, $\rho_R = 0.75$, 0.5 and 0.25 in rows 1, 2 and 3, respectively, and $\ell_R = 3$, 2 and 1 in columns 1, 2 and 3, respectively. In each group of 9 boxes, the first three are for $n_1 = n_2 = 10$ and $k_{\mathcal{I}} = 2$, the next three are for $n_1 = n_2 = 25$ and $k_{\mathcal{I}} = 2$, and the last three are for $n_1 = n_2 = 25$ and $k_{\mathcal{I}} = 1$. In each group of three boxes, the data are transformed by $\widehat{Q}^{-1}$ (first box), T^{-1} (second box), or untransformed (third box).

(SVM) classifier with a linear kernel (`svmtrain` in Matlab), and the logistic classifier applied to the Partial Least Squares (PLS) projection of the data (here data refers to any of the four versions, transformed or not, of the data); see Delaigle and Hall (2012b) and Section A.3.3 in the **Supplementary Material** for more details about the logistic classifier, and see Delaigle and Hall (2012a) for properties of PLS in the functional context.

Boxplots summarizing the results of our simulations, in the same settings as the centroid classifier, are shown in Figures 6 to 11 of Section A.3.5 in the **Supplementary Material**. From these figures we can see that the results obtained with these two classifiers are very similar to those with the centroid classifier. In other words, inverting by T^{-1} usually did not improve the results, and in general, inverting by the transform $Q^{-1}_{\widehat{\theta}_{\rm CV}}$, chosen by crossvalidation from the data, either improved the results significantly compared to using the data Y_{ij} or $T^{-1}Y_{ij}$, or, when the latter worked well, transforming the data by $Q^{-1}_{\widehat{\theta}_{\rm CV}}$ did not degrade performance much.

As allready noted, the best transform to apply generally depends on the particular classifier. However, an attractive aspect of our methodology is that the suggested inversion, $Q^{-1}_{\widehat{\theta}_{\rm CV}}$, is chosen to minimize a crossvalidation estimator of classification error. Therefore our approach is very flexible, since in a general setting it approximates the inverse transform

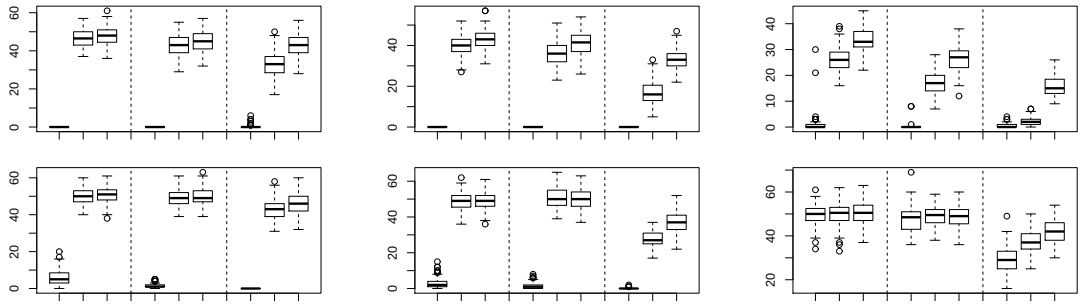

Figure 8: Boxplots of percentage of misclassified observations calculated from 100 simulated samples from models (c) (row 1) and (d) (rows 2) when R_r is of the form at (4.7), with $\theta_{M,r_j} = \theta_M + 2 \cdot [2\cos(r_j/2)]$ and $\theta_M = 30$, 20 and 10 in columns 1, 2 and 3, respectively. In each group of 9 boxes, the first three are for $n_1 = n_2 = 10$ and $k_{\mathcal{I}} = 2$, the next three are for $n_1 = n_2 = 25$ and $k_{\mathcal{I}} = 2$, and the last three are for $n_1 = n_2 = 25$ and $k_{\mathcal{I}} = 1$. In each group of three boxes, the data are transformed by $\widehat{Q}^{-1}$ (first box), T^{-1} (second box), or untransformed (third box).

that optimizes classification.

4.5 Empirical example

We have access to data from a small experiment involving long range infrared light detection and ranging (Lidar) methods. Briefly, the idea is to discriminate between two types of aerosols that have been emitted and are to be detected by Lidar, those that are biological in nature and those that are non-biological. There are 29 curves available to us, with $n_1 = 15$ non-biological and $n_2 = 14$ biological signals.

The process involves a signal or waveform sent out in a series of bursts, and received Lidar data were observed. Some of the bursts were sent before the aerosol was released, and these were used to background-correct the received signal after the aerosol was released. For each sample, the data used here are the background-corrected received signals for a burst, 19 wavelengths and 250 backscatter time points. In our illustrative analysis we followed the procedure described below for 20 bursts collected almost simultaneously in the middle of the release period and then averaged over the bursts before classification. Thus, in our notation, Y_{ij} consists of the two-dimensional collection of background-corrected received signals over the wavelengths and the backscatter time points for the i^{th} sample within the j^{th} aerosol class. This observed data is the convolution of a true signal, the Lidar response function for a delta-pulse transmitter, with the transmitted signal. If we write $\mathcal{G}_{ijw}(t)$ for this true signal

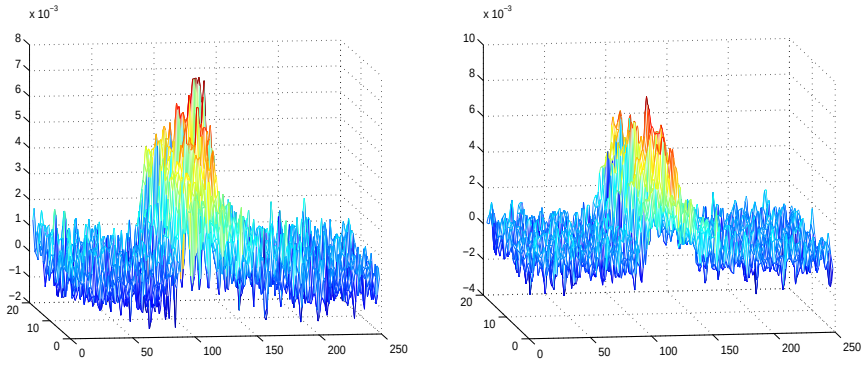

Figure 9: Plotted are two background corrected received data curves from population 1 (left) and population 2 (right), averaged over the 20 bursts, across wavelength and the backscatter spectral range.

for wavelength w at backscatter time point t, $R_{ijw}(t)$ for the background-corrected received signal, and $\mathcal{T}_w(t)$ for the transmitted signal, then, using an integral approximation to the discrete convolution, the signal we observe is

$$R_{ijw}(t) = \int_0^t \mathcal{G}_{ijw}(t-v)\mathcal{T}_w(v)\,dv + \kappa_{ijw}(t),$$

where $\xi_{ijw}(t)$ has mean zero. If we define $\mathcal{N}_{jw}(t) = E\{\mathcal{G}_{ijw}(t)\}$, $\mathcal{M}_{jw}(t) = \int_0^t \mathcal{N}_{jw}(t-v)\mathcal{T}_w(v)\,dv$ and $\mathcal{P}_{ijw}(t) = \int_0^t \{\mathcal{G}_{ijw}(t-v) - N_{jw}(t-v)\}\,\mathcal{T}_{ijw}(v)\,dv + \kappa_{ijw}(t)$, then we have that the observed data are given by $R_{ijw}(t) = \mathcal{M}_{jw}(t) + \mathcal{P}_{ijw}(t)$, where $\mathcal{P}_{ijw}(t)$ has mean zero. In our notation, Y_{ij}, μ_j, $T\mu_j$ and ϵ_{ij} are the collection of $R_{ijw}(t)$, $\mathcal{N}_{jw}(t)$, $\mathcal{M}_{jw}(t)$ and $\mathcal{P}_{ijw}(t)$ over the wavelengths and backscatter ranges, respectively, but averaged across 20 bursts. It is readily observed that the transformation $T\mu_j$ is linear. Two typical observed average curves for each population are given in Figure 9.

We considered three approaches. The first simply used the observed data Y_{ij}. The second was our method applied to the $Q^{-1}_{\widehat{\theta}_{CV}} Y_{ij}$, where Q_θ had the form at (4.4). In the third, for each burst and wavelength, we deconvolved to estimate $\mathcal{G}_{ijw}(t)$ using the Wiener-Helstrom method described by Warren, et al. (2008), and averaged over the bursts. In each case, since we could not generate new data, we estimated the misclassification error rate (i.e. misclassification percentage) by crossvalidation. In other words, as in the case of the procedure described in Section 2.3.3, we built the classifier from all but one of the 29 curves, classified that curve in one of the two populations (non-biological or biological), and averaged the results over all 29 curves.

For the centroid classifier, the crossvalidation estimator of misclassification error rate was

34.5% for the first approach based on non-transformed data, 24.1% for our crossvalidation based inversion approach, and 34.5% for the third approach based on inversion of T. For the SVM and logistic regression classifiers, the estimator of misclassification error rate was 37.9% (SVM) or 27.6% (logistic) when the classifier was based on non-transformed data, 17.2% (SVM) or 21% (logistic) when the classifier was based on our crossvalidation based inversion method, and 58.6% (SVM) or 31% (logistic) when the classifier was based on inversion of T. For all three classifiers, the reduction in misclassification error rate obtained by our crossvalidation-based data inversion illustrates the significant improvement that can be obtained by inverting the data through a data-driven transform chosen to minimize an estimator of classification error.

Acknowledgments

Carroll's research was supported by a grant from the National Cancer Institute (R37-CA057030) and in part by Award Number KUS-CI-016-04, made by King Abdullah University of Science and Technology (KAUST) and by the National Science Foundation (DMS-0914951). Delaigle's research was supported by grants and a Queen Elizabeth II Fellowship from the Australian Research Council, and Hall's research was supported by a Federation Fellowship, a Laureate Fellowship, and grants from the Australian Research Council.

References

Besag, J. (1986), "On the Statistical-Analysis of Dirty Pictures", *Journal of the Royal Statistical Society, Series B*, 48, 259–302.

Cannon, M. (1976), "Blind Deconvolution of Spatially Invariant Image Blurs with Phase," *IEEE Transactions on Acoustics, Speech and Signal Processing*, 24, 58–63.

Cannon, T. M. and Hunt, B. R. (1981), "Image Processing by Computer," *Scientific American*, 245, 214–225.

Carasso, A. S. (2001), "Direct Blind Deconvolution," *SIAM Journal on Applied Mathematics*, 61, 1980–2007.

Cressie N. and Kornak J. (2003), "Spatial Statistics in the Presence of Location Error with an Application to Remote Sensing of the Environment," *Statistical Science*, 18, 436–456.

Crosilla, F., Visintini, D. and Sepic, F. (2007), "An Automatic Classification and Robust Segmentation Procedure of Spatial Objects," *Statistical Methods and Applications*, 15, 329–341.

Dass, S. C. and Nair, V.N. (2003), "Edge Detection, Spatial Smoothing, and Image Reconstruction with Partially Observed Multivariate Data," *Journal of the American*

Statistical Association, 98, 77–89.

Delaigle, A. and Hall, P. (2012a), "Methodology and Theory for Partial Least Squares Applied to Functional Data," *Annals of Statistics*, 40, 322–352

Delaigle, A. and Hall, P. (2012b), "Componentwise Classification and Clustering of Functional Data," *Biometrika*, doi:10.1093/biomet/ass003. To appear.

Donoho, D. L. (1994), "Statistical Estimation and Optimal Recovery", *Annals of Statistics*, 22, 238–270.

Figueiredo, M. A. T. and Nowak, R. D. (2003), "An EM Algorithm for Wavelet-Based Image Restoration," *IEEE Transactions on Image Processing*, 12, 906–916.

Friedman, J. H. (1996), "Another Approach to Polychotomous Classification," Manuscript. `http://www-stat.stanford.edu/~jhf/ftp/poly.pdf`

Galatsanos, N. P., Mesarović, V. Z., Molina, R., Katsaggelos, A. K. and Mateos, J. (2002), "Hyperparameter Estimation in Image Restoration Problems with Partially Known Blurs," *Optical Engineering*, 41, 1845–1854.

Hall, P. (1990), "Optimal Convergence Rates in Signal Recovery," *Annals of Probability*, 18, 887–900.

Hall, P. and Qiu, P. (2007a), "Blind Deconvolution and Deblurring in Image Analysis," *Statistica Sinica*, 17, 1483–1509.

Hall, P. and Qiu, P. (2007b), "Nonparametric Estimation of a Point Spread Function in Multivariate Problems," *Annals of Statistics*, 35, 1512–1534.

Huang H.C. and Cressie, N. (2000), "Deterministic/Stochastic Wavelet Decomposition for Recovery of Signal from Noisy Data," *Technometrics*, 42, 262–276.

Huang, X. F. and Qiu, P. (2010), "Blind Deconvolution for Jump-Preserving Curve Estimation," *Mathematical Problems in Engineering*, Article No. 350849, doi 10.1155/2010/350849 (electronic).

James, G. and Hastie, T. (2001), "Functional Linear Discriminant Analysis for Irregularly Sampled Curves," *Journal of the Royal Statistical Society, Series B*, 63, 533–550.

Johnstone, I. M. (1990), "Speed of Estimation in Positron Emission Tomography and Related Inverse Problems," *Annals of Statistics*, 18, 251–280.

Joshi, M. V. and Chaudhuri, S. (2005), "Joint Blind Restoration and Surface Recovery in Photometric Stereo," *Journal of the Optical Society of America*, A22, 1066–1076.

Klein R. and Press, S. J. (1992), "Adaptive Bayesian Classification of Spatial Data," *Journal of the American Statistical Association*, 87, 844–851.

Kundur, D. and Hatzinakos, D. (1998), "A Novel Blind Deconvolution Scheme for Image Restoration Using Recursive Filtering," *IEEE Transactions on Signal Processing*, 46, 375–389.

McLachlan, G.J., Do, K.A. and Ambroise, C. (2004), *Analyzing Microarray Gene Expression Data*, Wiley, Hoboken, NJ.

Mukherjee, P. S. and Qiu, P. (2011), "3-D Image Denoising by Local Smoothing and Nonparametric Regression," *Technometrics*, 53, 196–208.

Popescu, D. C. and Hellicar, A. D. (2010), "Point Spread Function Estimation for a Terahertz Imaging System," *EURASIP Journal on Advances in Signal Processing*, Article No. 575817, doi 10.1155/2010/575817 (electronic).

Qiu, P. (2005), *Image Processing and Jump Regression Analysis*, New York: John Wiley and Sons.

Qiu, P. (2007), "Jump Surface Estimation, Edge Detection, and Image Restoration," *Journal of the American Statistical Association*, 102, 745–756.

Qiu, P. (2008), "A Nonparametric Procedure for Blind Image Deblurring," *Computational Statistics and Data Analysis*, 52, 4828–4841.

Shi, T. and Cressie, N. (2007), "Global Statistical Analysis of MISR Aerosol Data: a Massive Data Product From NASA's Terra Satellite," *Environmetrics*, 18, 665–680.

Shin, H. (2008), "An Extension of Fisher's Discriminant Analysis for Stochastic Processes," *Journal of Multivariate Analysis*, 99, 1191–1216.

Warren, R. E., Vanderbeek, R. G., Ben-David, A. and Ahl, J. L. (2008), "Simultaneous Estimation of Aerosol Cloud Concentration and Spectral Backscatter from Multiple-Wavelength Lidar Data," *Applied Optics*, 47, 4309–4320.