Iterative R&R (Rotation and Remarginalization) for Detecting Targets in Spectral Imagery

James Theiler[1] and Christopher X. Ren[2]

[1]Los Alamos National Laboratory, Los Alamos, NM
[2]Earthrise Media, Oakland, CA

ABSTRACT

We explore some variants of "Gaussianization" for characterizing the distribution of background pixels in multispectral and hyperspectral imagery, and then use this characterization to develop algorithms for target detection. We consider two very different problems – anomalous change detection and gas-phase plume detection – as ways to explore the applicability of Gaussianization for remote sensing image analysis.

One variant is an extension of the Gaussianization concept to non-Gaussian reference distributions, and in particular, we show that using the multivariate t as the reference distribution often leads to better target detection performance. Since we are no longer, strictly speaking, *Gauss*-ianizing, we call the method iterative rotation and remarginalization.

In our scheme, the remarginalization is achieved with a parametric transformation function that is built up from a linear basis of (hard or soft) hinge functions, which provide explicitly differentiable and enforcably monotonic remarginalization functions. An efficient knot-pruning strategy enables rapid training of these functions.

Also, for remote sensing imagery with many spectral channels, we have found it advantageous to pre-whiten the data with axes aligned to principal components, and then selectively to Gaussianize only the top principal components, treating the lower-variance directions as "already Gaussian." This provides a computationally faster and empirically more effective Gaussianization for spectral imagery.

Keywords: remote sensing, target detection, multispectral imagery, machine learning, density estimation, Gaussianization, remarginalization

In addition, it would be very desirable to develop an improved, but still easily calculated, PDF model that can describe EC as well as non-EC distributions. – Steven Adler-Golden[1]

1. INTRODUCTION

Gaussianization is a recently-developed approach[2–6] for estimating the density of a non-Gaussian distribution from data that is sampled from the distribution. The main idea is to find a transformation that maps the sample data to an approximately Gaussian distribution. The density, for any given point in the original distribution, is then given by the determinant of the transformation's Jacobian at that point, multiplied by the density of the Gaussian for the transformed data. In particular, if $\mathcal{T} : \mathbb{R}^d \to \mathbb{R}^d$ is an invertible transformation function that maps a point $\mathbf{x}$ to a point $\mathbf{y}$, where the $\mathbf{x}$ points are drawn from a (typically unknown) distribution $p_{\mathbf{x}}$, and the $\mathbf{y}$ points are distributed as a known reference distribution $p_{\mathbf{y}}$; then we can write

$$p_{\mathbf{x}}(\mathbf{x}) = p_{\mathbf{y}}(\mathbf{y}) \left| \frac{d\mathbf{y}}{d\mathbf{x}} \right| = p_{\mathbf{y}}(\mathcal{T}(\mathbf{x})) \left| \frac{d\mathcal{T}(\mathbf{x})}{d\mathbf{x}} \right| = p_{\mathbf{y}}(\mathcal{T}(\mathbf{x}))\mathcal{J}(\mathbf{x}) \tag{1}$$

where $\mathcal{J} = |d\mathcal{T}/d\mathbf{x}|$ is the absolute value of the determinant of the $d \times d$ Jacobian matrix of the transform $\mathcal{T}$. Instead of trying to infer $p_{\mathbf{x}}$ directly from data, the Gaussianization process tries to infer the transform $\mathcal{T}$ that leads to $p_{\mathbf{y}}$ being Gaussian. With $p_{\mathbf{y}}$ and $\mathcal{T}$ known, then the above formula gives $p_{\mathbf{x}}$.

There is, however, no formal reason that $p_{\mathbf{y}}$ must be Gaussian. In principle, any reference distribution can work, but it should be one for which $p_{\mathbf{y}}$ can be readily evaluated, and for which the marginal distribution obtained

by an arbitrary projection of $p_{\mathbf{y}}$ can be readily evaluated. The Gaussian fits this bill, but an alternative that we found effective is the multivariate t distribution. Like the Gaussian, the multivariate t is elliptically-contoured (EC), and the projection along any axis is a t distribution. The multivariate t is fatter tailed than the Gaussian, and often provides a better fit to hyperspectral data.[7]

1.1. Background estimation and target detection

The characterization of the background distribution is a crucial step for many analysis tasks in multispectral remote sensing imagery, and particularly for the detection of targets in that imagery.[8] Given a model for the background distribution, one can directly use this model for anomaly detection, anomalous change detection, and a broad range of target detection problems.

1.2. Anomaly and anomalous change detection

In the anomaly detection problem, for example, the aim is to determine, from a collection of samples (usually pixels*) $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, which of these samples are the least consistent with the underlying background distribution; that is, which $\mathbf{x}$'s exhibit the smallest values of $p_{\text{bkg}}(\mathbf{x})$. Thus $\mathcal{A}(\mathbf{x}) = 1/p_{\text{bkg}}(\mathbf{x})$ is a natural choice for an anomaly detection function.

For the anomalous change detection problem, one has two images of the same scene, taken at different times, under conditions that are inevitably different, and possibly even with different sensors. Given these two images, the aim is to find pixels that correspond to interesting changes in the scene. That is: find the pixel pairs (*i.e.*, the corresponding pixels from the two images) that are mutually unusual (compared to the the other pixel pairs in the image pair), but without regard to whether the pixels are individually unusual.

The approach we take, first proposed in Ref. 9, can be expressed in terms of three distributions. If $\mathbf{x}_A$ and $\mathbf{x}_B$ are corresponding pixels in images A and B respectively, and if $\mathbf{x}_{AB} = \begin{bmatrix} \mathbf{x}_A \\ \mathbf{x}_B \end{bmatrix}$ is a vector corresponding to the pixel pair, then we can write the anomalousness of change as

$$\mathcal{A}(x) = \frac{p_{AB}(\mathbf{x}_{AB})}{p_A(\mathbf{x}_A)p_B(\mathbf{x}_B)} \tag{2}$$

While anomaly and target detection scenarios required the estimation of a single background distribution, the anomalous change expression in Eq. (2) has three distributions to be learned. In principle, one only needs to learn p_{AB}, the distribution for the stacked pixels, because p_A and p_B are just lower-dimensional projections of p_{AB}, but we have not found an efficient way to perform that projection (nominally, it would require an integration at each point of interest). For our work here, we will simply estimate each of these three distributions separately.

We remark that Padrón-Hidalgo *et al.*[6] also employ Gaussianization for change detection, but with an entirely different approach that seeks pixels in the second image that are anomalous with respect to the distribution learned from that first image: that is, $\mathcal{A}(x) = 1/p_A(x_B)$. This has the advantage that it only requires the estimation of a single distribution (instead of three), but the disadvantage that it is insensitive to changes due to movement of objects in a scene. It furthermore is not robust to inevitable differences in conditions (*e.g.*, due to calibration, illumination, view-angle, *etc*) that are pervasive over the scene, and does not even make sense if different sensors are employed.

1.3. Plume detection

For general target detection problems, where the target has a known spectral signature, we can write $\boldsymbol{\xi}(\mathbf{x})$ as the effect on the spectrum of the observed pixel of having a target present in a pixel $\mathbf{x}$. The optimal detector in this case is given by the likelihood ratio:

$$\mathcal{L}(\mathbf{x}) = \frac{p_{\text{tgt}}(\mathbf{x})}{p_{\text{bkg}}(\mathbf{x})} = \frac{p_{\text{bkg}}(\boldsymbol{\xi}^{-1}(\mathbf{x}))}{p_{\text{bkg}}(\mathbf{x})} \left| \frac{d\boldsymbol{\xi}}{d\mathbf{x}} \right|. \tag{3}$$

*We will, for the sake of concreteness, often refer to samples as pixels, but the formulations we describe could also take samples to be pixel patches that include spatial and textural information as well as spectral.

For an additive target, for example: $\boldsymbol{\xi}(\mathbf{x}) = \mathbf{x} + a\mathbf{t}$, where a is the strength of the target and $\mathbf{t}$ is the target signature. For this case, $|d\boldsymbol{\xi}/d\mathbf{x}| = 1$ and $\mathcal{L}(\mathbf{x}) = p_{\text{bkg}}(\mathbf{x} - a\mathbf{t})/p_{\text{bkg}}(\mathbf{x})$.

For the plume detection problem, the aim is to find gas-phase plumes – which can for instance be indicative of impending volcanic eruptions,[10,11] pollution,[12,13] or greenhouse gas emission[14–16] – in a scene observed by a remote, possibly satellite-based,[17,18] hyperspectral sensor. In contrast to solid targets, the presence of plume only slightly perturbs the observed spectrum, but although the magnitude of the effect is small, the nature of the perturbation is quite precisely known, and for images with many spectral channels, even fairly weak plumes can be detected. In the visible regime, the effect is primarily absorptive, and we can write[19]

$$\boldsymbol{\xi}(\mathbf{x}) = \exp(-aT)\mathbf{x} \tag{4}$$

where $T = \text{diag}(\mathbf{t})$ is a diagonal matrix whose diagonal entries correspond to the absorption spectrum $\mathbf{t}$ for the gas, and a is a measure of plume concentration. This leads to

$$\mathcal{L}(a, \mathbf{x}) = \frac{p_{\text{tgt}}(\mathbf{x})}{p_{\text{bkg}}(\mathbf{x})} = \frac{p_{\text{bkg}}\left(\exp(aT)\mathbf{x}\right)\, \exp(a\tau)}{p_{\text{bkg}}(\mathbf{x})}, \tag{5}$$

where $\tau = \text{trace}(T)$, which is the sum of the components of $\mathbf{t}$. Eq. (5) is the so-called clairvoyant detector; it has nice optimality properties, but is rarely useful in practice because it depends on the plume strength a which is generally not known. This is the *composite hypothesis testing* problem, and a considerable variety of approaches have been developed to deal with this ambiguity;[20–25] here, we will follow Ref. 26 and use a fixed value $a = n/\sqrt{\boldsymbol{\mu}' T' R^{-1} T \boldsymbol{\mu}}$, which corresponds to n "sigmas" of separation between on-plume and off-plume pixels (we use $n = 3$ sigmas). Here $\boldsymbol{\mu}$ and R correspond to the mean and covariance of the background pixels.

Various approximations to Eq. (5) can be made; and combining those approximations with closed-form expressions for the background distribution (specifically, by assuming Gaussian or multivariate t-distributed[27] or log-normal[28] backgrounds). Here, we relax the assumption that we know the form of p_{bkg}, and use IR&R to estimate it.

2. IR&R: ITERATIVE ROTATION AND REMARGINALIZATION

Our iterative rotation and remarginalization algorithm follows the standard Gaussianization procedure, but does not necessarily use a Gaussian. Details of the training algorithm are shown in Algorithm 1. The returned rotation matrices R_m and scalar squashing functions H_{mk} comprise a description of the transformation function $\mathcal{T}$. Specifically,

$$\mathcal{T}(\mathbf{x}) = R_M H_M(R_{M-1} H_{M-1}(\cdots R_2 H_2(R_1 H_1(\mathbf{x})) \cdots)) \tag{6}$$

where $H_m : \mathbb{R}^d \to \mathbb{R}^d$ is a vector-valued function whose components are given by $H_{mk} : \mathbb{R}^d \to \mathbb{R}$ for $k = 1, \ldots, d$. It has previously been noted[3] that this successive application of linear rotation and nonlinear squashing has the structure of a neural network, and in our previous work[5] we explicitly trained a neural network to learn $\mathcal{T}(\mathbf{x})$ directly. As this can often be done with many fewer than M layers in the trained neural network, some efficiency is gained by this method. But it is important to recognize that the form of the expression for $\mathcal{T}(\mathbf{x})$ in Eq. (6) has some advantages of its own. Most notably, the determinant of the Jacobian of $\mathcal{T}(\mathbf{x})$ can be efficiently computed,* and this permits the density estimator in Eq. (1) to be efficiently evaluated. Details are shown in Algorithm 2.

2.1. Remarginalization

The remarginalization step corresponds to separate (and independent) one-dimensional Gaussianizations along each of the d dimensions of the data. At the end of this step, the marginal distributions of the the transformed points will be Gaussian along each of these dimensions. Although one-dimensional Gaussianization is in principle fairly straightforward, the Appendix describes an approach for doing this in a way that is relatively efficient and robust.

*The determinant of the rotation matrices is always one, and because the squashing functions H_m have diagonal Jacobians, their determinants are just products of their diagonal elements. Combining these with the fact that $|AB| = |A| \cdot |B|$ for matrices A, B, we can compute the determinant of the Jacobian of $\mathcal{T}(x)$ without ever explicitly computing the Jacobian itself.

Algorithm 1 Iterative Rotation and Remarginalization (Training)

Require: $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, with $\mathbf{x}_n \in \mathbb{R}^d$
  Initialize $\mathbf{y}_n \leftarrow \mathbf{x}_n$, for all $n = 1, \ldots, N$
  for $m = 1, 2, \ldots, M$ **do** ▷ Iterate until $\mathbf{y}_n$'s are, by some measure, sufficiently Gaussian
    $R_m \leftarrow$ RandomRotationMatrix
    $\mathbf{y}_n \leftarrow R_m \mathbf{y}_n$, for all n ▷ Rotate data
    for $k = 1, \ldots, d$ **do** ▷ Remarginalize each component k separately
      Let $z_n \leftarrow \mathbf{y}_n^{(k)}$ be kth component of $\mathbf{y}_n$, for all n
      $H_{mk} \leftarrow$ GetSquashingFunction($\{z_1, \ldots, z_N\}$)
      Let $\mathbf{y}_n^{(k)} \leftarrow H_{mk}(z_n)$, for all n ▷ Reset the k'th component of $\mathbf{y}_n$
  Return: R_m, H_{mk} for $m = 1, \ldots, M$ and $k = 1, \ldots, d$

  function GetSquashingFunction($\{z_1, \ldots, z_N\}$)
    Find monotonic function h so that set $\{h(z_1), \ldots, h(z_N)\}$
    is distributed as a Gaussian or multivariate t distribution ▷ See Appendix for details
    return h

Algorithm 2 Iterative Rotation and Remarginalization (Applying)

Require: $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, with $\mathbf{x}_n \in \mathbb{R}^d$ ▷ May be different from X in Algorithm 1
Require: R_m, H_{mk} for $m = 1, \ldots, M$ and $k = 1, \ldots, d$ ▷ Obtained from Algorithm 1
  Initialize $\mathbf{y}_n \leftarrow \mathbf{x}_n$, for all $n = 1, \ldots, N$ ▷ Vector $\mathbf{y}_n$ corresponds to transformed value of $\mathbf{x}_n$
  Initialize $s_n \leftarrow 1$ for $n = 1, \ldots, N$ ▷ Scalar s_n corresponds to determinant of Jacobian at $\mathbf{x}_n$
  for $m = 1, 2, \ldots, M$ **do**
    $\mathbf{y}_n \leftarrow R_m \mathbf{y}_n$, for all n ▷ Rotate data
    for $k = 1, \ldots, d$ **do** ▷ Remarginalize each component k separately
      Let $z_n \leftarrow \mathbf{y}_n^{(k)}$ be kth component of $\mathbf{y}_n$, for all n
      Let $\mathbf{y}_n^{(k)} \leftarrow H_{mk}(z_n)$, for all n ▷ Reset the k'th component of $\mathbf{y}_n$
      Update $s_n \leftarrow s_n \times \frac{dH_{mk}}{dz}\big|_{z_n}$ for all n
  Compute $\rho_n = |s_n| \cdot \mathcal{G}(\mathbf{y}_n)$ for all n ▷ with $\mathcal{G}$ the Gaussian (or multivariate t) distribution function
  Return: ρ_n for all n ▷ ρ_n is estimate of density at $\mathbf{x}_n$

2.2. Initial Rotation

If these d dimensions were statistically independent of each other, then the result of this transformation would not only be Gaussian along the specified dimensions, but would be a d-dimensional multivariate Gaussian. And we would be done: because $p_{\mathbf{y}}$ would then be known, Eq. (1) could be used to accurately model the distribution $p_{\mathbf{x}}$ of the original untransformed data.

This motivates the use of whitening as a pre-processing stage. Here, the data with mean $\boldsymbol{\mu}$ and covariance R are rescaled

$$\mathbf{x} \leftarrow R^{-1/2}(\mathbf{x} - \boldsymbol{\mu}) \tag{7}$$

so that the resulting data has zero mean and unit covariance matrix. The axes in whitened data are not typically independent, but they *are* uncorrelated. Note that matrices do not have unique square roots, and the $R^{-1/2}$ term in the above equation is ambiguous as written. The square root that we recommend is the one that aligns the principal components to the whitened axis. We can achieve this by using as our inverse square root the matrix $R^{-1/2} := D^{-1/2}U^{\mathsf{T}}$ where D and U are, respectively, the diagonal and unitary components in the singular value decomposition of the symmetric positive-definite covariance matrix $R = UDU^{\mathsf{T}}$. Here, $D^{-1/2}$ is computed by taking the inverse positive square root of each of the diagonal entries.

With this as our whitening operator, the axes of our whitened data will be aligned with the PC directions. Now, remarginalization along each of the PC directions is very much along the lines of an approach invented by Adler-Golden[1] for anomaly detection in heavy-tailed backgrounds. In that work, the data were not explicitly remarginalized; but the effect of component-wise remarginalization was achieved by fitting a separate model to each PC coordinate. A second motivation for using PCs is that many of them are Gaussian already; as Adler-Golden writes[1]:

> The leading (low-numbered) PCs capture the largest variations in the dataset; they tend to show non-Gaussian statistics, characterized by heavy tails. In contrast, the trailing (high-numbered) PCs tend to be noise-dominated and observe Gaussian statistics, with much shorter tails.

That the most non-Gaussian directions tend to be correlated with highest-variance directions had been observed earlier,[29,30] but Adler-Golden's algorithm provided a principled way to exploit that observation to obtain improved anomaly detection performance using all of the PCs.* More recently, Tidhar and Rotman[31] took this approach with a multidimensional histogram-based nonparametric density model for the first several PCs, and an assumption of Gaussianity for the later PCs. We will also take this approach, iteratively rotating and remarginalizing only the top PCs.

3. EXPERIMENTS

Since IR&R is a nonparametric approach that is designed to be able to characterize more or less arbitrary background distributions, we will be using real hyperspectral datasets as training data. To avoid ambiguities in ground truth, however, the performance of this training process will be evaluated using targets that are artificially implanted into these images. For the plume detection problems, we'll take the two-histogram[32] or matched-pair[33] approach. This involves making a copy of our hyperspectral data and implanting target in every pixel of the copy.

Then the detector (that was trained only on the original data) is applied to both the original and the implanted data. The detection rate (DR) is the fraction of pixels in the implanted data sets for which targets are detected. We adjust the threshold on the detector so that the detection rate is fixed (we use DR=0.5), and then compute the false alarm rate (FAR@DR=0.5) as the fraction of pixels in the original image for which the detector indicated target presence. (Because some of those false alarms might correspond to real targets, the quantity we compute is a conservative estimate of the true false alarm rate and in any case provides a measure that can be used to compare different detectors.) The anomalous change detection problem will use a similar

*The subspace RX (SSRX) algorithm[30] makes indirect use of this observation, by projecting out the low-numbered (high variance) PCS and only using the high-numbered PCs.

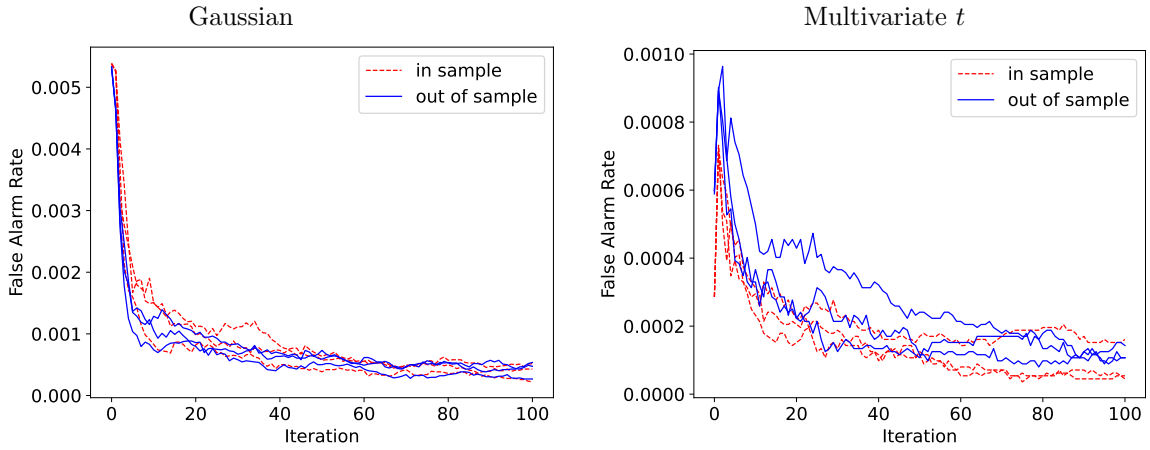

Figure 1. False alarm rate (FAR@DR=0.5) as a function of iteration for IR&R applied to the anomalous change detection problem. Since rotations are random, three different runs are shown. The left panel uses remarginalization to a Gaussian and the right panel uses a multivariate t distribution with $\nu = 3.5$ degrees of freedom. In both cases, IR&R provided considerable reduction in the false alarm rate.

strategy, but the "implanted" targets will be provided by existing pixels chosen randomly from elsewhere in the image.[34]

For all of the experiments, we cut the image into stripes that are 10 pixels wide, and then train on even-numbered stripes while testing on odd-numbered stripes; we use stripes instead of just a random partition of the pixels to reduce pixel adjacency effects between the testing and training sets. From the in-sample and out-of-sample ROC curves, we produce the FAR@DR=0.5 statistic, corresponding to the false alarm rate at the threshold that produces a 50% detection rate.

3.1. Anomalous change detection

Our anomalous change detection (ACD) experiment begins with two images from the RIT Blind Test dataset.[35,36] Two hyperspectral images were taken of an area around Cooke City, MT, USA. We do not have ground truth for the changes between these images, but we evaluate performance using a pixel scrambling scheme,[34] which simulates anomalous changes by randomly moving pixels to new locations. Although the data is hyperspectral, we reduce the dimension, using canonical correlation analysis, to ten channels for each image. These channels are linear combinations of the spectral channels that produce the highest correlation between the images; higher correlation is advantageous because it makes the rare non-correlated pixels (*i.e.*, the anomalous changes) stand out more.[34] This statistic is plotted against iteration number in Fig. 1. Compared to the assumption of a Gaussian background distribution (corresponding to the 0th iteration), we see striking improvement with Gaussianization. To some extent, what the algorithm is "learning" is that the distribution is better modeled by a multivariate t. Indeed, the performance for multivariate t is much better (observe the y-axis values), and iterations of R&R provide further improvement.

3.2. Plume detection

We begin with data provided by JPL from overflights of the AVIRIS-NG sensor.[37] After making a copy of the image, and implanting a uniform plume of methane gas over the whole image, we transform the data to principle component coordinates and scale those coordinates so that the resulting data is whitened.

Although the data is high dimensional (with 186 spectral channels), we only apply the rotation and remarginalization to the top $d = 10$ dimensions. What we observe in Fig. 2 is that IR&R has virtually no effect on plume detection performance. After a hundred iterations toward a Gaussian distribution, we see that the false alarm rate changes by about a percent. We see that using a multivariate t distribution instead of a Gaussian as a starting point, we get about a five percent gain in performance. But the effect of the IR&R from those starting

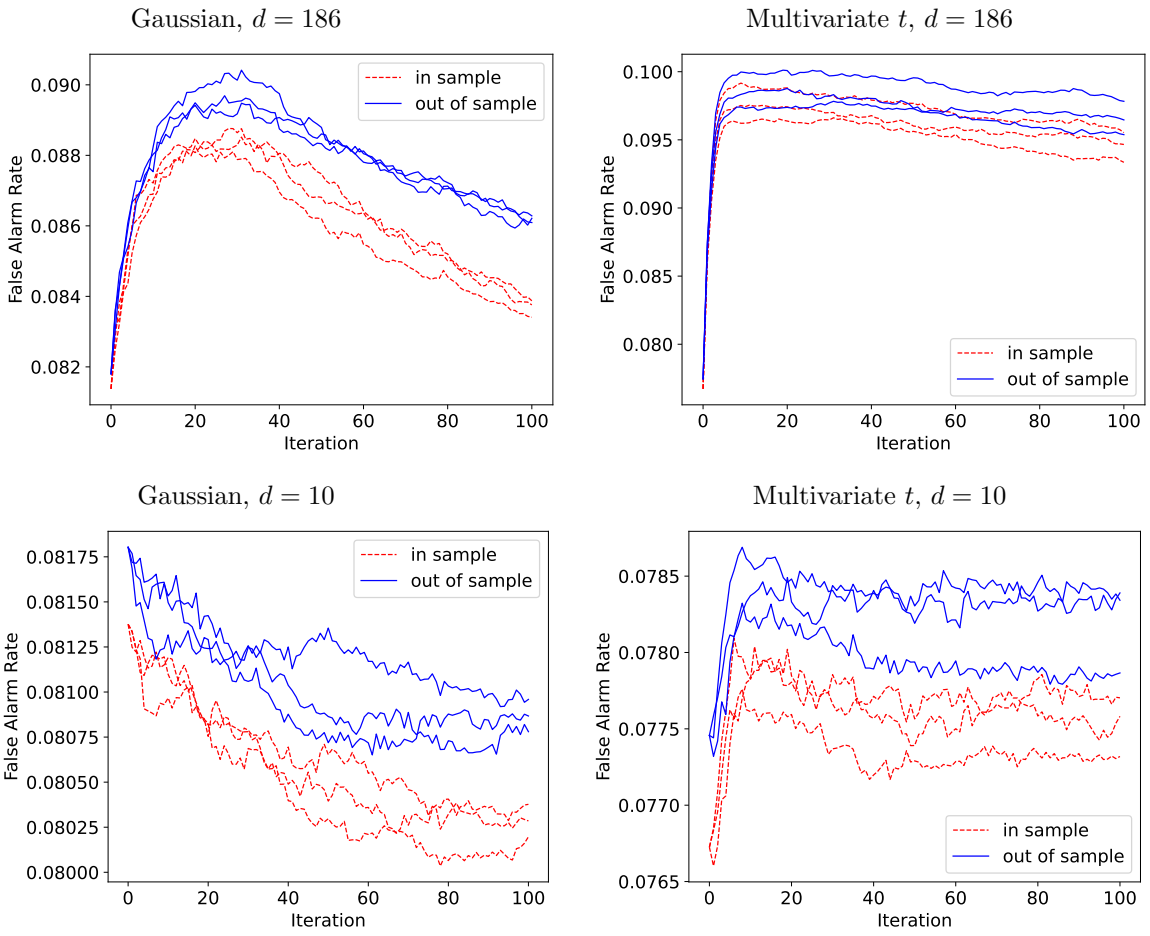

Figure 2. False alarm rate (FAR@DR=0.5) as a function of iteration for IR&R applied to the plume detection problem. Since rotations are random, three different runs are shown. The left panels use remarginalization to a Gaussian and the right panel uses a multivariate t distribution with $\nu = 3.5$ degrees of freedom. The top panels use all $d = 186$ spectral channels, while the bottom panels only transform the top $d = 10$ principal components. Overall, IR&R has very little effect on the false alarm rate, though working with reduced dimensionality appears to be an advantage.

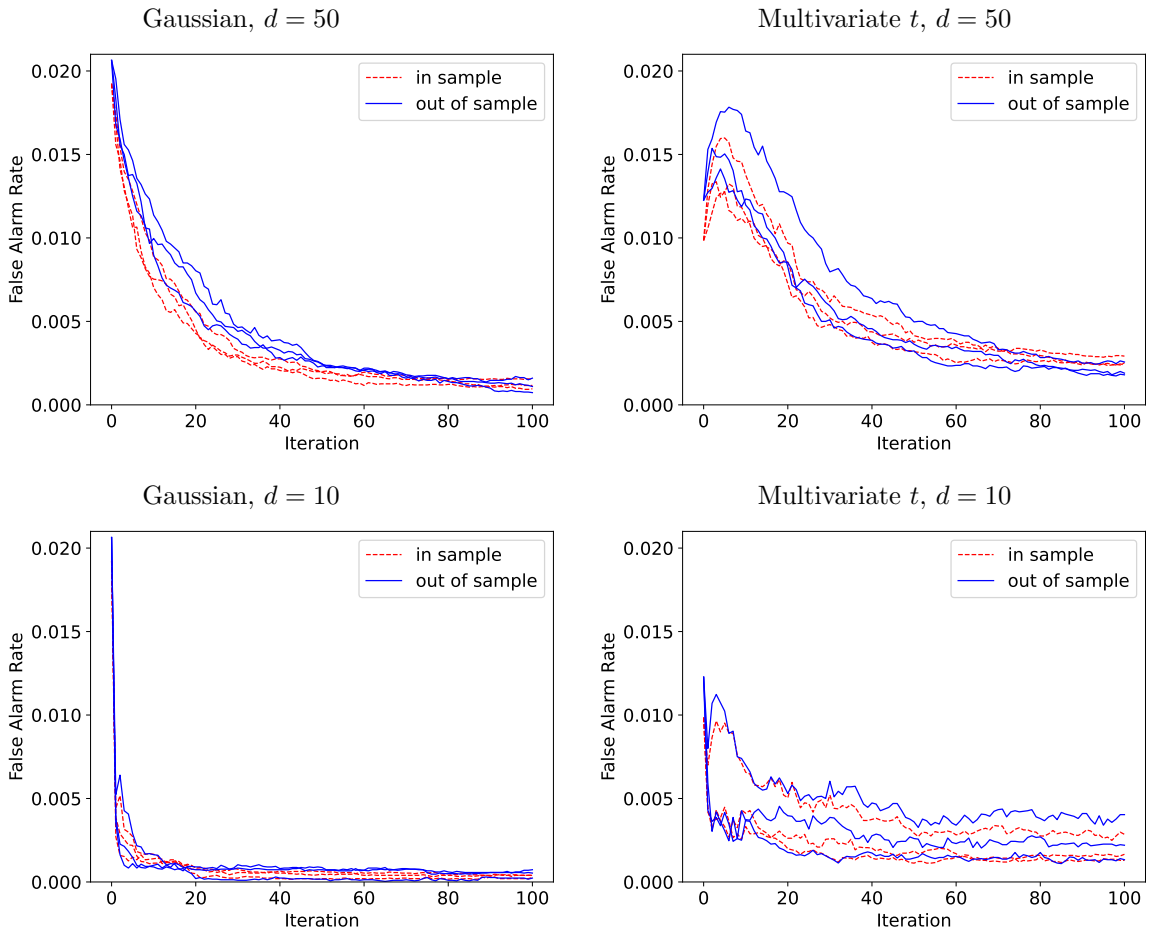

Figure 3. False alarm rate (FAR@DR=0.5) as a function of iteration for IR&R applied to the plume detection problem, using different data. Since rotations are random, three different runs are shown. The left panels use remarginalization to a Gaussian and the right panel uses a multivariate t distribution with $\nu = 3.5$ degrees of freedom. The top panels use all $d = 50$ spectral channels, while the bottom panels only transform the top $d = 10$ principal components. Here, the advantage of reduced dimensionality is evident both for Gaussian and multivariate t. Interestingly, the Gaussian IR&R actually outperforms the multivariate t IR&R, even though the initial multivariate t model was better than the Gaussian.

points is less than a percent. And for the multivariate t starting point, the IR&R makes it worse (but, again, by a very small amount).

One of the reasons plume detection works so well for hyperspectral imagery is that the gas-phase absorption spectrum typically involves multiple very narrow lines, whereas the background spectrum of solid materials tends to show reflectances that vary relatively smoothly with wavelength. Geometrically, we can imagine the background pixels as a cloud of points in a d-dimensional spectral space (with one axis per spectral channel); this cloud has some directions with a lot of variation and other directions that are quite "thin" and do not vary much at all. The high-variance directions, which correspond to lower-numbered principal components (PCs), describe the main variability in the solid background; the target spectrum (*i.e.*, the absorption spectrum for the gas species of interest) tends to be orthogonal to these high-variance directions, and in directions with relatively low variance in the background – this is what enables even small deviations from that background (*i.e.*, weak plumes) to be detectable.

The test this hypothesis – that the reason IR&R performed so poorly is that the target signature is buried in the high-numbered (*i.e.*, already Gaussian) PCs – we performed a second experiment. Actually, the experiment

was the same, but we used different data. We used data from the SHARE 2012 experiment[38] designed for solid sub-pixel targets.[39] We treated the spectrum of the solid (green panel) targets as if it were a gas absorption spectrum and implanted a simulated gas plume with this signature. The results of this experiment are shown in Fig. 3. Here, we see that modeling the non-Gaussian (and even the non-multivariate t) nature of the background that is enabled with IR&R leads to substantially better detection performance; indeed, the false alarm rate is reduced by roughly a factor of ten. For this data, we also observe that the Gaussian IR&R outperforms the multivariate t IR&R, even though the initial multivariate t model was better than the Gaussian. A potential explanation for this is that although the higher-variance directions are better modeled by a fat-tailed multivariate t, the low-variance directions may be more Gaussian. A hybrid of Gaussian and multivariate t (very similar to what Adler-Golden suggested in Ref. [1]) may lead to better performance with fewer iterations.

4. CONCLUSION

That hyperspectral backgrounds are manifestly complicated is not a problem but an opportunity. Given the success of detection algorithms that employ simple background models – notably,Gaussian or multivariate t distributions – we can reasonably expect improved performance from detectors that are based on more flexible background models. To some extent, we have seen that in this study, which looked at two quite different detection algorithms. For the anomalous change detection problem, the closer modeling of the background provided by the IR&R algorithm led to lower false alarm rates. For the gas-phase plume detection problem, the results were more nuanced (which is to say: not as good). We did not improve the performance at all on a real plume spectrum implanted in a real hyperspectral background. But we did see improvement when that spectrum was modified to be more aligned with the background spectra. It remains to be seen whether this observation can be leveraged into a variant algorithm that is more successful with gas-phase plumes.

An observation worth making here is that IR&R aims to model the "whole" distribution, and is specifically designed to minimize the Kullback-Liebler distance between the model and the underlying distribution. But what we measure is false alarm rate, and that depends only on the tail of the distribution. This explains why even in-sample false alarm rate can increase with iterations of IR&R.

Indeed, an argument against this approach is that it violates Vapnik's dictum[40] that one should not solve a more difficult problem (in this case, density estimation) as an intermediate step in solving a simpler problem (target detection). In the jargon of machine learning, this is a distinction between generative and discriminative modeling. Where some methods (such as Refs. [33, 41, 42]) attempt to directly discriminate target from background, we are here achieving that discrimination only after obtaining a full density model.

On the other hand, IR&R has an appealing property that is not available in, for instance, kernel-based or neuron-based learning algorithms; and that is that it *begins* with a Gaussian (or multivariate t) distribution. There are many algorithms in spectral data analysis, and in spectral target detection in particular, that are known to work fairly well, even if not necessarily optimally, and that are derived by assuming the background distribution is Gaussian or multivariate t. With IR&R, we essentially use those methods as our starting point. We may (as in the case of anomalous change detection) or may not (as in the case of gas-phase plume detection) improve upon those results, but we know that we can at least *achieve* them.

Acknowledgments

We are grateful to the Los Alamos Laboratory Directed Research and Development program and the National Aeronautics and Space Administration for supporting this work. Thanks also to the Rochester Institute of Technology and to the NASA Jet Propulsion Laboratory for making publicly available the data that was used in this project. Finally and mostly, thank you to Steven Adler-Golden, whose clear insights and humble demeanor have long been an inspiration.

APPENDIX A. REMARGINALIZATION: A BASIS SET FOR MONOTONIC FUNCTION ESTIMATION USING PAIRS OF HINGE FUNCTIONS

The remarginalization step corresponds to Gaussianizing in one dimension, and this is in principle fairly straightforward. Let $f(x)$ be a one-dimensional density function, and then $F(x) = \int_{-\infty}^{x} f(z)\, dz$ is the corresponding *cumulative* density function (CDF). If we write $\Phi(x)$ as the CDF for a Gaussian*, then $\Phi^{-1}F$ is a monotonically increasing (and therefor invertible) function that Gaussianizes the data. In particular, if the data set $\{x_1, \ldots, x_N\}$ is drawn from the original distribution, then the set $\{\Phi^{-1}(F(x_1)), \ldots, \Phi^{-1}(F(x_N))\}$ exhibits a Gaussian distribution.

In practice, of course, we do not know $f(x)$; what we have as our input is the dataset $\{x_1, \ldots, x_N\}$ that is presumed to be drawn from $f(x)$. There are many ways to estimate $f(x)$ from data , but what we *really* want is not $f(x)$, *per se*, but the function $H(x) = \Phi^{-1}(F(x))$. In this appendix, we will describe our approach for estimating $H(x)$ from the input dataset. Since $H(x)$ will be constrained to be continuous and monotonic, a sorted list of input arguments will lead to a sorted list of output values; the effect of H then is to locally "stretch and squeeze" the input arguments; we will often refer to this remarginalization function as a *squash*ing function.

We will treat the fitting of $H(x)$ as a regression problem. We begin by sorting the input values; without loss of generality, write $\{x_1, \ldots, x_N\}$ is the input set, but with the property that $x_1 \leq \cdots \leq x_N$. Now, associate $H(x_n)$ with $y_n = \Phi^{-1}(a_n)$ where $a_n = (n - \frac{1}{2})/N$ uniformly fills the interval $[0, 1]$. (We could also take y_n to be the n'th value in a sorted list of N numbers that are randomly drawn from the reference distribution.)

We will fit $y_n \sim H(x_n)$ by taking H from a parametric class of monotonic functions. Ideally, this class should exhibit the following properties:

1. $H(x)$ is guaranteed to be monotonic increasing (preferably with slope strictly greater than zero)
2. $H(x)$ is flexible enough to fit arbitrary 1-d distributions
3. $H(x)$ has a functional form that makes it easy (meaning fast) to fit to the data.
4. $H(x)$ is (everywhere, or at least almost everywhere) smooth and differentiable.
5. $H(x)$ is easy to differentiate (we need this because we will be computing Jacobians)
6. $H(x)$ is easy to invert. This is not needed for our target detection application; but it *is* helpful if we want to use $\mathcal{T}^{-1}$ to generate new samples from the original $p_{\mathbf{x}}$ distribution.
7. The identity function ($H(x) = x$) should be particularly easy to fit, since that's what we are converging to as the transformed data become more like the reference distribution.

A.1. Piecewise linear (hard) hinges

Our initial scheme[5] used a three-segment piecewise linear function. Here, we have extended that to arbitrary number of segments, by treating the squashing function as a linear combination of hinge functions. The base hinge function is given by

$$h(x) = \begin{cases} 0 & \text{for } x \leq 0 \\ x & \text{for } x \geq 0 \end{cases} \tag{8}$$

This base hinge has a "knot" point at $x = 0$, but we can write $h(x - c)$ as a hinge function with a knot at $x = c$. If we have K distinct knot points: $c_0 < c_1 < \cdots < c_{K-1}$, then we can write an arbitrary $K+1$-segment piecewise linear function as

$$H(x) = a + bx + \sum_{k=0}^{K-1} w_k h(x - c_k) \tag{9}$$

*We can write $\Phi(x)$ more broadly as the CDF of the reference distribution; it is typically Gaussian, but in much of our work, we consider a t distribution instead.

There are $K+1$ distinct regions along the x-axis: $x < c_0$, $c_0 < x < c_1$, ..., $c_{K-1} < x$. The slope of $H(x)$ is constant in each of those regions*, and is given by b, $b+w_0$, $b+w_0+w_1$, ..., $b+w_0+\cdots+w_{K-1}$. If we want to ensure monotonicity, then we need the following $K+1$ conditions, corresponding to non-negative slope in each of tho4 $K+1$ regions.

$$0 \le b \tag{10}$$
$$0 \le b + w_0 \tag{11}$$
$$\vdots$$
$$0 \le b + w_0 + \cdots + w_{K-1} \tag{12}$$

These constraints are convex, and therefore they can be efficiently (though not very conveniently) enforced. Our implementation uses an equivalent formulation that allows a more convenient enforcement of monotonicity. Write:

$$G_1(x) = x - h(x - c_0) \tag{13}$$
$$G_2(x) = h(x - c_0) - h(x - c_1) \tag{14}$$
$$\vdots$$
$$G_k(x) = h(x - c_{k-2}) - h(x - c_{k-1}) \quad \text{for } k = 2, \ldots, K \tag{15}$$
$$\vdots$$
$$G_{K+1} = h(x - c_{K-1}) \tag{16}$$

and then the squashing map has the form

$$H(x) = g_0 + \sum_{k=1}^{K+1} g_k G_k(x). \tag{17}$$

In this case, each of the basis functions $G_k(x)$ are by construction monotonic, and therefore we can enforce monotonicity in $H(x)$ simply by requiring $g_k \ge 0$ for $k = 1, \ldots, K+1$. See Fig. 4.

As a further re-parameterization, we construct $d_k = c_{k+1} - c_k$ for $k = 0, \ldots, K-2$, and consider the list $c_0, d_0, \ldots, d_{K-2}$ in place of $c_0, \ldots, c_{K-1}$. We do this so that we can impose $d_k > 0$ as a way of ensuring the the knots satisfy $c_0 < c_1 < \cdots < c_{K-1}$. This is useful, because some of our fitting algorithms allow the knots to be free parameters as well as the coefficients.

Note that these two parameterizations ($[a, b, c_0, \ldots, c_{K-1}, w_0, \ldots, w_{K-1}]$ and $[g_0, \ldots, g_{K+1}, c_0, d_0, \ldots, d_{K-2}]$) are equivalent, with:

$$g_0 = a \tag{18}$$
$$g_1 = b \tag{19}$$
$$g_2 = b + w_0 \tag{20}$$
$$\vdots$$
$$g_{K+1} = b + w_0 + \cdots + w_{K-1} \tag{21}$$
$$d_0 = c_1 - c_0 \tag{22}$$
$$\vdots$$
$$d_{K-2} = c_{K-1} - c_{K-2} \tag{23}$$

*The slope is not well defined at the knot points themselves; it's not clear, however, that this is truly a problem. As long as a slope in the range $[b + w_0 + \cdots + w_{k-1}, b + w_0 + \cdots + w_k]$ is assigned to the point at x=c_k, then I think things will be fine. We'll see later how smoothing the hinge function avoids this problem (assuming it *is* a problem).

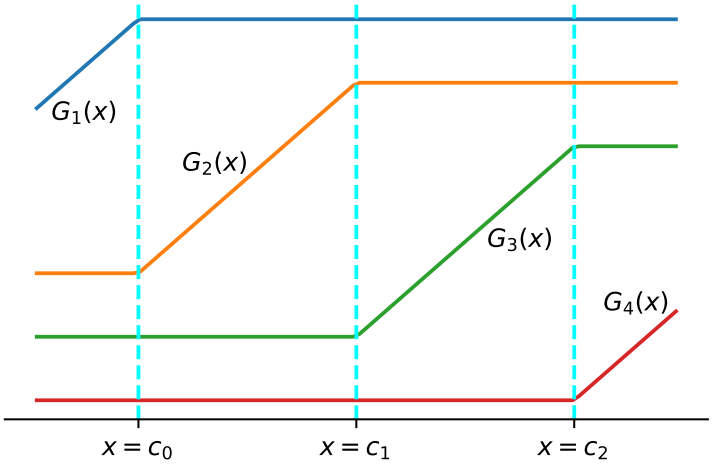

Figure 4. Four basis functions are shown (vertically offset for clarity), corresponding to Eqs. (13-16), for the case with $K = 3$ knot points. Linear combinations of these basis function can be used to generate four-segment piecewise linear functions. And positive linear combinations will yield monotonically increasing piecewise linear functions.

and

$$a = g_0 \tag{24}$$
$$b = g_1 \tag{25}$$
$$c_1 = c_0 + d_0 \tag{26}$$
$$\vdots$$
$$c_{K-1} = c_0 + d_0 + \cdots + d_{K-2} \tag{27}$$
$$w_0 = g_2 - g_1 \tag{28}$$
$$\vdots$$
$$w_{K-1} = g_{K+1} - g_K \tag{29}$$

Indeed, when we actually *evaluate* Eq. (17), we do that by first converting the $(\mathbf{g}, c, \mathbf{d})$ parameters into $(a, b, \mathbf{c}, \mathbf{w})$ parameters, and evaluating Eq. (9). This avoids duplication in the evaluation of the hinge functions.

For the algorithms that fix the knot locations before fitting the linear coefficients, we put the knots at midpoints between adjacent values in the data; that way the derivative will be unambiguous, at least for the training data. (For out of sample data, it will be ambiguous only on a measure zero set of the data.)

A.2. Inverting the squashing function

If $\mathcal{T}(\mathbf{x})$ is the transform that converts data from the original distribution into a Gaussian, then the inverse $\mathcal{T}^{-1}(\mathbf{x})$ can take Gaussian data as input and produce samples from the original distribution. One step in this full inversion is to invert the squashing function $H(x)$. An advantage of this piecewise-linear characterization of the squashing function is that it is readily inverted. Since the inverse is also a monotonically increasing piecewise linear function, we can write an explicit expression for the parameters associated with the inverse. So if $(\mathbf{g}, c, \mathbf{d})$

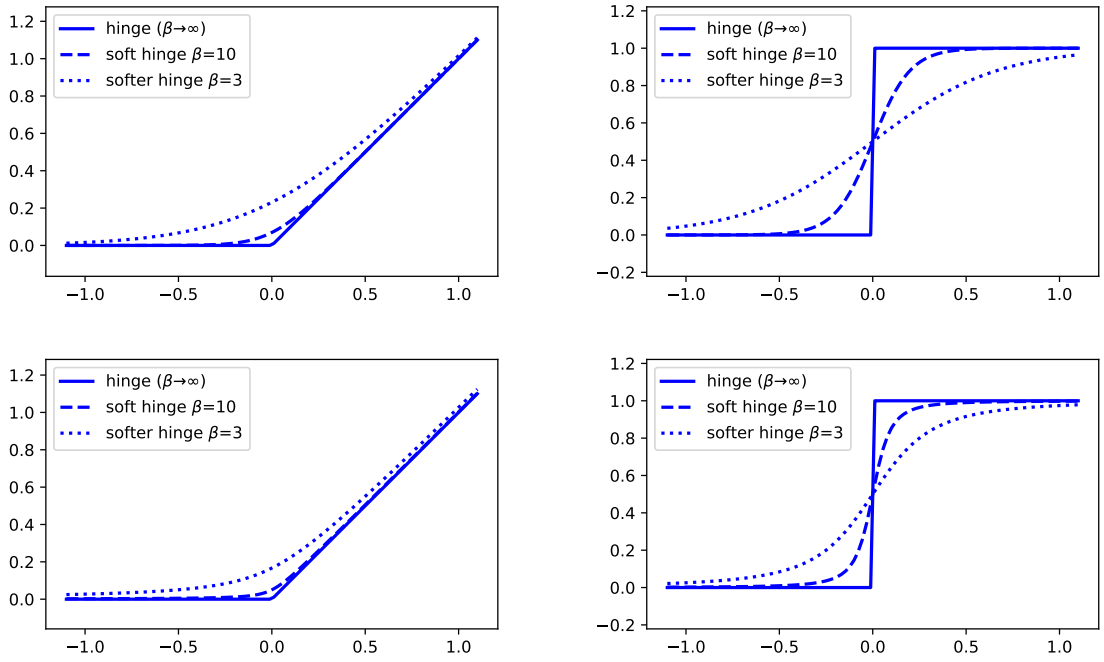

Figure 5. Hinge and soft hinge functions (left panels), and their derivatives (right panels). The top panels use the log-exponential formulation in Eq. (34) and Eq. (35); the bottom panels use the square-root-square formulation in Eq. (36) and Eq. (37).

are the parameters associated with $H(x)$, then the parameters for $H^{-1}(x)$ are given by $(\mathbf{g}', c', \mathbf{d}')$, where:

$$g'_0 = -g_0/g_1 \tag{30}$$
$$g'_k = 1/g_k, \quad \text{for } k = 1, \ldots, K+1 \tag{31}$$
$$c'_0 = g_0 + g_1 c_0 \tag{32}$$
$$d'_k = g_{k+2} d_k, \quad \text{for } k = 0, \ldots, K-2 \tag{33}$$

A.3. Smoothing the squashing function

The piecewise linear functions are flexible (with enough knots, could fit any monotonic function to arbitrary accuracy) and continuous, but the first derivative is not continuous, and the first derivative an important part of the density estimation. We can address this problem by replacing the hinge functions with "soft" hinge functions. Here, a parameter β is introduced, and:

$$h(\beta; x) = \frac{1}{\beta} \log\left(1 + \exp(\beta x)\right) \tag{34}$$
$$\frac{d}{dx} h(\beta; x) = \frac{\exp(\beta x)}{1 + \exp(\beta x)} \tag{35}$$

In this formulation, we see that $1/\beta$ is a kind of "smoothness" distance; for $|x| \gg 1/\beta$ – that is, for x far away from the hinge point, the smoothed hinge function approaches the simple hinge function. For large β, this smoothness distance is small, the the smoothed hinge function is well approximated by the simple hinge. See Fig. 5. However, the smoothed hinge is infinitely differentiable over the full domain. Furthermore, we have (as we did for the simple hinge functions), that differences $h(\beta; x - c_0) - h(\beta; x - c_1)$ are strictly monotonically increasing functions as long as $c_0 < c_1$.

A.3.1. Alternative soft hinge function based on the square root of the square

The log-exponential formulation in Eq. (34) and Eq. (35) is a standard approach for softening the hinge function*, and it is indeed a convenient and adjustable approach. We have implemented an alternative softening, however, based on the square root of a square; specifically:

$$h(\beta; x) = \frac{1}{2\beta}\left[\beta x + \sqrt{1 + (\beta x)^2}\right] \tag{36}$$

$$\frac{d}{dx}h(\beta; x) = \frac{1}{2}\left[1 + \frac{\beta x}{\sqrt{1 + (\beta x)^2}}\right] \tag{37}$$

We have found that this formulation is somewhat faster to evaluate than the expressions in Eq. (34) and Eq. (35). Furthermore, we do not need to worry about overflow of the exponential with large arguments†

A.4. Hybrid hard and smooth fitting for squashing functions

Since the smooth squashing functions are approximations of the hard hinges, one approach is to use the hard hinges (which are much cheaper to compute, and much less prone to numerical instability) for fitting the coefficients (the **g**'s) and then using those coefficients with smooth hinges in the final squashing function. [That doesn't sound very convincing, but in practice it really works.]

A.5. Fixed knots, and knot pruning

Identifying the best parameters is a problem in nonlinear (and nonconvex) optimization, and can be computationally expensive, especially since every iteration requires a separate fit for each dimension of the data.

If the knot locations are fixed, however, the remaining free parameters are linear coefficients, and can be optimized using standard least squares fitting. In our implementation, the knots are chosen so that the number of samples between each adjacent pair of knots is roughly the same.

We have also implemented a two-step process, in which a relatively larger number of initial knot locations are identified, a "quick fit" is made using all of the knots, and then knots are pruned out, one at a time, until the desired number of knots is reached. A new fit is made with these remaining knots. The pruning scheme removes points according to how much the fit changes if the knot is removed.

Specifically, consider the piecewise linear function that is produced by connecting adjacent knots with line segments. For each knot (x_n, y_n), we compute the area between that piecewise linear function and the piecewise linear function that is obtained if that knot point were removed. To do this, we compute for each knot the value $\hat{y}_n$ corresponding to the y-value at x_n that would be obtained if the knot with index n were removed. The area of the difference between the two is the area of a triangle with height given by the difference between $\hat{y}_n$ and y_n, and base given by the distance $x_{n+1} - x_{n-1}$. That is (see Fig. 6):

$$\hat{y}_n = \frac{y_{n+1}(x_n - x_{n-1}) + y_{n-1}(x_{n+1} - x_n)}{x_{n+1} - x_{n-1}} \tag{38}$$

$$A_n = \frac{1}{2}\left|y_n - \hat{y}_n\right|(x_{n+1} - x_{n-1}) \tag{39}$$

The algorithm for pruning knots finds the knot with the smallest value of A_n, removes that knot from the list, and then recomputes $\hat{y}$ and A values, continuing to remove knots from the list until the desired number of knots remain.

The quicker of the "quick fits" takes advantage of the fact that the (x, y) data is monotonic; so from sorted lists of x and y values, the y associated with each knot can be immediately identified. Indeed, this is a reasonable fit in its own right, but it is particularly useful as an initial fit both to provide a starting point for the nonlinear fit, and to provide the quick fit to a larger set of knot points, and from which a pruned set of knot points can be derived for use in a full least squares fit.

*For example, the function is called "softplus" in the torch package.[43]

†In practice, this is addressed in the log-exponential by comparing the argument to a threshold, and if it is larger than the threshold, then using the asymptotic value instead of explicitly evaluating the exponential and then the logarithm.

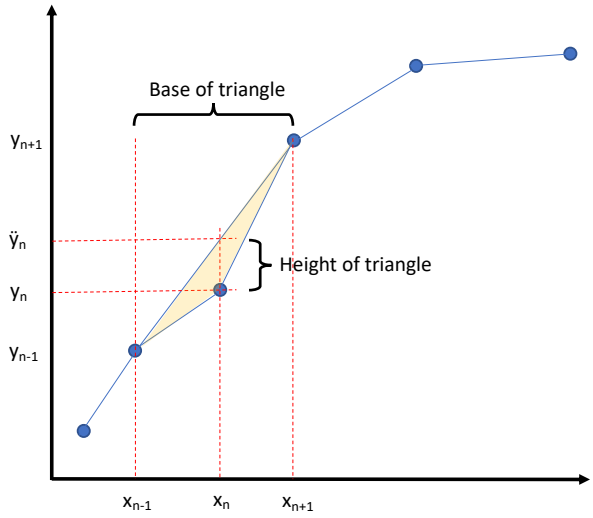

Figure 6. For each knot at (x_n, y_n), we assess its importance by the area of the triangle that is created if the knot were removed. The knot associated with the smallest area (*i.e.*, the knot most co-linear with its neighbors) is removed. We begin with many more knots than we want, and remove them one at a time until a desired number of knots is reached.

A.6. Fractional squashing

Previously,[5] we argued that we could avoid some instabilities in the iterative R&R process by using a "fractional" squashing. Given data pairs $(x_1, y_1), \ldots, (x_N, y_N)$ obtained by separately sorting the x's and the y's, the standard (or "full" or "non-fractional") fit seeks a function $H(x)$, for instance given by Eq. (17), that minimizes $\sum_n \|y_n - H(x_n)\|^2$. The idea of fractional (or partial) fitting is to minimize a function of the form $\sum_n \|\widetilde{y}_n - H(x_n)\|^2$, where $\widetilde{y}_n = (1-f)x_n + f y_n$.

Here $f = 1$ corresponds to standard fitting, and if the the aim were to do the best fit in a single step, then $f = 1$ would be the appropriate choice. But IR&R is iterative, and as long as each iteration nudges the distribution "towards" the reference (usually Gaussian) distribution, then progress is being made. The convergence towards the reference distribution depends on which rotations are made at each iteration; by using $f < 1$, we make this dependence less strong. Although our early experience with three-segment piecewise linear fits[5] suggested improved performance with f as small as 0.3 to 0.5, more recent experiments with our hinge pair basis suggest that larger f is better, and in some cases $f = 1$ appears optimal. In the experiments reported here, we used f=0.9.

REFERENCES

1. S. M. Adler-Golden, "Improved hyperspectral anomaly detection in heavy-tailed backgrounds," in *Proc. 1st Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, IEEE, 2009.
2. S. S. Chen and R. A. Gopinath, "Gaussianization," *Advances in Neural Information Processing Systems (NIPS)* **13**, pp. 423–429, 2000.
3. V. Laparra, G. Camps-Valls, and J. Malo, "Iterative Gaussianization: from ICA to random rotations," *IEEE Trans. Neural Networks* **22**(4), pp. 537–549, 2011.
4. C. Meng, Y. Song, J. Song, and S. Ermon, "Gaussianization flows," *Proc. 23rd International Conference on Artificial Intelligence and Statistics* **108**, pp. 4336–4345, 2020.
5. J. Theiler and C. X. Ren, "Distilled Gaussianization," *Proc. SPIE* **11843**, p. 118430G, 2021.
6. J. A. Padrón-Hidalgo, V. Laparra, and G. Camps-Valls, "Unsupervised anomaly and change detection with multivariate Gaussianization," *IEEE Transactions on Geoscience and Remote Sensing* **60**, 2021.

7. D. Manolakis, D. Marden, J. Kerekes, and G. Shaw, "On the statistics of hyperspectral imaging data," *Proc. SPIE* **4381**, pp. 308–316, 2001.
8. S. Matteoli, M. Diani, and J. Theiler, "An overview of background modeling for detection of targets and anomalies in hyperspectral remotely sensed imagery," *J. Selected Topics in Applied Earth Observations and Remote Sensing (JSTARS)* **7**, pp. 2317–2336, 2014.
9. J. Theiler and S. Perkins, "Proposed framework for anomalous change detection," *ICML Workshop on Machine Learning Algorithms for Surveillance and Event Detection* , pp. 7–14, 2006.
10. S. P. Love, F. Goff, D. Counce, C. Siebe, and H. Delgado, "Passive infrared spectroscopy of the eruption plume at Popocatépetl v olcano, Mexico," *Nature* **396**, pp. 563–567, 1998.
11. S. P. Love, F. Goff, S. C. Schmidt, D. Counce, D. Pettit, B. W. Christenson, and C. Siebe, "Passive infrared spectroscopic remote sensing of volcanic gases: Ground-based studies at White Island and Ruapehu, New Zealand, and Popocatépetl, Mexico," in *Remote Sensing of Active Volcanism*, P. Mouginis-Mark, J. Crisp, and J. Fink, eds., *Geophysical Monograph* **116**, pp. 117–138, American Geophysical Union, Washington, D.C., 2000.
12. A. K. Mebust, A. R. Russell, R. C. Hudman, L. C. Valin, and R. C. Cohen, "Characterization of wildfire NO_X emissions using MODIS fire radiative power and OMI tropospheric NO_2 columns," *Atmospheric Chemistry and Physics* **11**, pp. 5839–5851, 2011.
13. K. N. Buckland, S. J. Young, E. R. Keim, B. R. Johnson, P. D. Johnson, and D. M. Tratt, "Tracking and quantification of gaseous chemical plumes from anthropogenic emission sources within the Los Angeles basin," *Remote Sensing of Environment* **201**, pp. 275–296, 2017.
14. D. R. Thompson, I. Leifer, H. Bovensmann, M. Eastwood, M. Fladeland, C. Frankenberg, K. Gerilowski, R. O. Green, S. Kratwurst, T. Krings, B. Luna, and A. K. Thorpe, "Real-time remote detection and measurement for airborne imaging spectroscopy: a case study with methane," *Atmospheric Measurement Techniques* **8**, pp. 4383–4397, 2015.
15. C. Frankenberg, A. K. Thorpe, D. R. Thompson, G. Hulley, E. A. Kort, N. Vance, J. Borchardt, T. Krings, K. Gerilowski, C. Sweeney, S. Conley, B. D. Bue, A. D. Aubrey, S. Hook, and R. O. Green, "Airborne methane remote measurements reveal heavy-tail flux distribution in four corners region," *Proc. National Academy of Sciences* **113**, pp. 9734–9739, 2016.
16. M. D. Foote, P. E. Dennison, A. K. Thorpe, D. R. Thompson, S. Jongaramrungruang, C. Frankenberg, and S. C. Joshi, "Fast and accurate retrieval of methane concentration from imaging spectrometer data using sparsity prior," *IEEE Trans. Geoscience and Remote Sensing* **58**, pp. 6480–6492, 2020.
17. J. Theiler, B. R. Foy, C. Safi, and S. P. Love, "Onboard cubesat data processing for hyperspectral detection of chemical plumes," *Proc. SPIE* **10644**, p. 1064405, 2018.
18. S. P. Love, L. A. Ott, J. Theiler, B. R. Foy, C. L. Safi, M. E. Dale, C. G. Peterson, A. A. Guthrie, N. A. Dallman, K. G. Boyd, P. S. Stein, J. A. Wren, M. C. Proicou, and M. K. Dubey, "High-resolution hyperspectral imaging of dilute gases from cubesat platforms," *American Geophysical Union, Fall Meeting* , pp. abstract #A41K–3107, 2018.
19. J. Theiler and A. Schaum, "Some closed-form expressions for absorptive plume detection," in *Proc. IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 1786–1789, 2020.
20. S. M. Kay, *Fundamentals of Statistical Signal Processing: Detection Theory*, vol. II, Prentice Hall, New Jersey, 1998.
21. E. L. Lehmann and J. P. Romano, *Testing Statistical Hypotheses*, Springer, New York, 2005.
22. J. Chen, "Penalized likelihood-ratio test for finite mixture models with multinomial observations," *Canadian Journal of Statistics* **26**, pp. 583–599, 1998.
23. A. Schaum, "Continuum fusion: a theory of inference, with applications to hyperspectral detection," *Optics Express* **18**, pp. 8171–8181, 2010.
24. P. Bajorski, "Min-max detection fusion for hyperspectral images," *Proc. 3rd IEEE Worskhop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)* , 2011.
25. J. Theiler, "Confusion and clairvoyance: some remarks on the composite hypothesis testing problem," *Proc. SPIE* **8390**, p. 839003, 2012.
26. J. Theiler, "Veritas: an admissible detector for targets of unknown strength," *Proc. SPIE* **11727**, 2021.

27. J. Theiler, "Absorptive weak plume detection on Gaussian and non-Gaussian background clutter," *J. Selected Topics in Applied Earth Observations and Remote Sensing (JSTARS)* **14**, pp. 6842–6854, 2021.
28. A. Schaum, "A uniformly most powerful detector of gas plumes against a cluttered background," *Remote Sensing of the Environment* **260**, p. 112443, 2021.
29. J. Theiler, B. R. Foy, and A. M. Fraser, "Characterizing non-Gaussian clutter and detecting weak gaseous plumes in hyperspectral imagery," *Proc. SPIE* **5806**, pp. 182–193, 2005.
30. A. Schaum, "Hyperspectral anomaly detection: Beyond RX," *Proc. SPIE* **6565**, p. 656502, 2007.
31. G. A. Tidhar and S. R. Rotman, "Target detection in inhomogeneous non-Gaussian hyperspectral data based on nonparametric density estimation," *Proc. SPIE* **8743**, p. 87431A, 2013.
32. M. Bar-Tal and S. R. Rotman, "Performance measurement in point target detection," *Infrared Physics & Technology* **37**, pp. 231–238, 1996.
33. J. Theiler, "Matched-pair machine learning," *Technometrics* **55**, pp. 536–547, 2013.
34. J. Theiler, "Quantitative comparison of quadratic covariance-based anomalous change detectors," *Applied Optics* **47**, pp. F12–F26, 2008.
35. D. Snyder, J. Kerekes, I. Fairweather, R. Crabtree, J. Shive, and S. Hager, "Development of a web-based application to evaluate target finding algorithms," *Proc. IEEE International Geoscience and Remote Sensing Symposium (IGARSS)* **2**, pp. 915–918, 2008.
36. Rochester Institute of Technology (RIT) Digital Imaging and Remote Sensing Laboratory, "Target detection blind test." `http://dirsapps.cis.rit.edu/blindtest/`. Accessed: July 2022.
37. NASA Jet Propulsion Laboratory, "Benchmark dataset for methane and carbon dioxide plumes." `https://avirisng.jpl.nasa.gov/benchmark_methane_carbon_dioxide.html`. Accessed: May 2022.
38. A. Giannandrea, N. Raqueno, D. W. Messinger, J. Faulring, J. P. Kerekes, J. van Aardt, K. Canham, S. Hagstrom, E. Ontiveros, A. Gerace, J. Kaufman, K. M. Vongsy, H. Griffith, B. D. Bartlett, E. Ientilucci, J. Meola, L. Scarff, and B. Daniel, "The SHARE 2012 data campaign," *Proc. SPIE* **8743**, p. 87430F, 2013.
39. J. P. Kerekes, K. Ludgate, A. Giannandrea, N. G. Raqueno, and D. S. Goldberg, "SHARE 2012: Subpixel detection and unmixing experiments," *Proc. SPIE* **8473**, p. 84730H, 2013.
40. V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, 2nd ed., 1999.
41. J. Theiler, "Transductive and matched-pair machine learning for difficult target detection problems," *Proc. SPIE* **9088**, p. 90880E, 2014.
42. A. Ziemann, M. Kucer, and J. Theiler, "A machine learning approach to hyperspectral detection of solid targets," *Proc. SPIE* **10644**, p. 1064404, 2018.
43. PyTorch, "Softplus." `https://pytorch.org/docs/stable/generated/torch.nn.Softplus.html`. Accessed: Aug 2021.