Matched-pair machine learning

James Theiler
Space Data Systems
Los Alamos National Laboratory

Abstract

Following an analogous distinction in statistical hypothesis testing, and motivated by chemical plume detection in hyperspectral imagery, we investigate machine learning algorithms where the training set is comprised of matched pairs. We find that even conventional classifiers exhibit improved performance when the input data has a matched-pair structure, and we develop an example of a "dipole" algorithm to directly exploit this structured input. In some scenarios, matched pairs can be generated from independent samples, with the effect of not only doubling the nominal size of the training set, but of providing the matched-pair structure that leads to better learning. The creation of matched pairs from a data set of interest also permits a kind of transductive learning which is found for the plume detection problem to exhibit improved performance. This paper has supplementary material online.
Keywords: Algorithms, Classification, Hyperspectral Imagery, Hypothesis Testing, Signal Detection, Structured Data

Technometrics **55** (2013) 536-547
`http://amstat.tandfonline.com/loi/tech`

1 Introduction

In perhaps the simplest formulation of machine learning (Vapnik, 1999; Hastie et al., 2001; Duda et al., 2001), one is given a training set of data samples $\mathbf{x} \in \mathbb{R}^d$ and associated labels $y \in \{-1, +1\}$, and the aim is to learn a function $f(\mathbf{x})$ that predicts the label y that is associated with $\mathbf{x}$. The purpose of this paper is to investigate a variant of that problem in which the training samples can be organized into matched pairs: $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2)\}$, with $y_1 \neq y_2$ and $\mathbf{x}_2$ dependent on $\mathbf{x}_1$.

An underlying assumption, in both the standard formulation and in this variant, is that the $\mathbf{x}$ values are drawn from distributions (a separate distribution for each label), which typically are not known. Indeed, if the distributions were known, one could immediately find the Bayes optimal function $f(\mathbf{x})$ for predicting labels from data. In the standard variant, $\mathbf{x}$ values are drawn independently from the distributions; in the matched-pair variant, however, the values are drawn in pairs, with one sample from each class. There is a natural analogy, developed further in Section 1.1, to hypothesis testing with matched pairs. Just as exploiting matched-pair structure in the hypothesis testing problem can lead to smaller p-values, exploiting matched-pair structure in the binary classification problem can lead to fewer misclassification errors.

This matched-pair structure is not always available for problems of interest, but one place where it can arise is with some signal detection problems. A specific example, and the motivation for this work, is described in Section 1.2. A more detailed formulation of the matched-pair approach is provided in Section 2. Three data sets are described in Section 3, along with a taxonomy of matched-pair structures. Section 4 applies both batch and online learning algorithms to these data sets, and Section 5 concludes.

1.1 Analogy: hypothesis testing

A common task in statistical hypothesis testing is to determine whether two separate sets of data samples arise from the same distribution. In one textbook example (Crawley, 2005), measurements of a biodiversity score are taken upstream and downstream of a sewage outfall. See Table 1. The mean downstream score is lower than the mean upstream score, but the

Table 1: An example of a statistical hypothesis testing problem: biodiversity scores downstream and upstream of a sewage outfall (Crawley, 2005).

Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Down	20	15	10	5	20	15	10	5	20	15	10	5	20	15	10	5
Up	23	16	10	4	22	15	12	7	21	16	11	5	22	14	10	6

difference is substantially less than the day-to-day variation in these measurements. The question is whether that difference is statistically significant.

If the upstream and downstream measurements are treated as independent samples, then a Student's t-test gives a statistically insignificant p-value of $p = 0.69$. If it is recognized that the upstream and downstream values are *not* independent of each other, but form matched pairs, then the matched-pair variant of the t-test can be invoked, and that gives $p = 0.008$, which *is* significant (Crawley, 2005).

This is a situation that arises frequently in statistical practice. Another well-known example (Box et al., 1978; Venables and Ripley, 2002) tests two materials used for children's shoes; each child got one shoe of each material, and the amount of wear was recorded. Here, the material-to-material difference is of interest, but it must be disentangled from the wide variations in how different children treat their shoes. These are situations where a matched-pair statistic is more appropriate, and more powerful, than the statistic that treats the data samples independently.

In machine learning, the task is a little different. Rather than asking whether two different states (*e.g.,* upstream and downstream) have significantly different effects, one seeks a classifier that distinguishes the states based on observations of the effects. For example, the task might be to infer from a given biodiversity score whether it was obtained from an upstream or a downstream location. In this case, the classifier would just be a threshold. More commonly, classification problems are based on multi-dimensional measurements with nontrivial boundaries between the classes. But if those classification problems have a matched-pair structure in the training data, then we will see that this structure can lead to better classification accuracy.

1.2 Motivation: finding weak signals

Of particular interest is in the detection of gas-phase chemical plumes in hyperspectral imagery, and in later sections (3.3 and 4.1.4), the problem will be described in greater detail. What is important about this problem is that the effect of a plume on a pixel spectrum is well understood; it suppresses radiation in wavelengths where the plume is absorptive, by an amount whose variation with wavelength is precisely known (Beer, 1852; Hayden et al., 1996; Foy et al., 2009). For wavelengths in the far-infrared, there are emissive effects as well (Manolakis, 2008). Identifying where a plume is in a hyperspectral image involves (very roughly speaking) finding those pixels whose radiance in the plume's absorptive wavelengths is smaller than would be expected, given the radiance in the other wavelengths.

But even when the effect of the plume is precisely known, the detection of plumes is still challenging because the background is cluttered in a way that is not, *a priori*, known at all. Explicitly characterizing this background with a high-dimensional probability distribution is difficult – and is a more general problem than the one we actually need to solve.

The traditional machine-learning approach to this kind of problem is to identify some pixels where the plume is present, some other pixels where the plume is known to be absent, and to infer an optimal boundary in spectral space (*i.e.,* the high-dimensional space of radiances at different wavelengths) that divides the two sets. Two problems with this approach are that 1) an adequately large and representative set of on-plume pixels is rarely available in practice, and 2) the domain knowledge we have about the effect of plumes on pixels is ignored. We can address both of these issues by creating artificial on-plume pixels from the off-plume pixels by applying the known plume absorption spectrum to them. This not only doubles the size of the training set, it produces a matched-pair structure within that training set which can improve the performance of classifiers that are trained with that data.

A practical consideration is that one may not know which pixels in an image truly are free of plume. If one were to plunge ahead and treat all the training pixels as off-plume, and create on-plume pixels from them, one could still produce a classifier. The performance of this classifier would presumably suffer from the contaminated samples, but for plumes that are rare and/or weak, we might expect the effect of this contamination to be small.

We can also use the matched-pair formalism to achieve a kind of machine learning that is related to the concept of "transductive inference" introduced by Vapnik (1999, p. 293). In this scheme, we train the classifier on a matched-pair set that treats the data of interest as nominally off-plume (even though the data may well contain pixels for which a plume is truly present), and creates artificial on-plume pixels from those nominally off-plume pixels. A classifier trained on these pixels is then applied back to the data of interest (*i.e.,* to those

nominally off-plume pixels) for the purpose of detecting the pixels for which plume is present. The scheme is transductive because it employs the data of interest in the classifier training, but (unlike in-sample learning) it does not have access to labels (*i.e.,* on-plume or off-plume) for that data. Indeed, the aim is to infer those labels.

2 Formulation

Let $p(\mathbf{x})$ be a probability density function over $\mathbf{x} \in \mathbb{R}^d$. Let the function $\boldsymbol{\xi} : \mathbb{R}^d \to \mathbb{R}^d$ represent a "treatment" that modifies $\mathbf{x}$. In the matched-pair problem, we know $\boldsymbol{\xi}$ but we do not know $p(\mathbf{x})$, and our aim is to find a function $f : \mathbb{R}^d \to \{-1, +1\}$ which distinguishes $\mathbf{x}$ from $\boldsymbol{\xi}(\mathbf{x})$ by classifying them into groups labeled -1 and $+1$. Thus, the function f will distinguish treated from untreated data samples; that is, it will *detect* those data samples to which the treatment has been applied.

For the hyperspectral plume detection problem, each pixel in the image has a vector-valued $\mathbf{x}$ corresponding to the spectrum of intensities measured at that pixel; the different scalar components of the vector $\mathbf{x}$ correspond to the intensities at different wavelengths. In this problem, $\boldsymbol{\xi}(\mathbf{x})$ represents the effect of a plume. If $\mathbf{x}$ is the spectrum of a pixel where plume is absent, the spectrum after a plume is introduced at that pixel will be given by $\boldsymbol{\xi}(\mathbf{x})$.

In the traditional machine-learning scenario, there is an unknown parent distribution $p(\mathbf{x}, y)$. Since $y \in \{-1, +1\}$ for binary classification, this amounts to two separate density functions over the input $\mathbf{x}$ – they are $p(\mathbf{x}, -1)$ and $p(\mathbf{x}, +1)$ – and a scalar probability for choosing from between them.

What is different in the matched-pair classification scenario is that there is only a single unknown density function – that is $p(\mathbf{x}, -1)$ – and a *known* process $\boldsymbol{\xi}$ that defines the other:[1]

$$p(\mathbf{x}, +1) = \left| \frac{\partial \boldsymbol{\xi}}{\partial \mathbf{x}} \right|^{-1} p(\boldsymbol{\xi}^{-1}(\mathbf{x}), -1). \tag{1}$$

Given density functions for the two classes, one can write down the Bayes optimal detector in terms of their ratio: $\mathcal{D}(\mathbf{x}) = p(\mathbf{x}, +1)/p(\mathbf{x}, -1)$. When $\mathcal{D}(\mathbf{x})$ is greater than a given threshold, $\mathbf{x}$ is predicted to have label $+1$, and when it is less than the threshold, then the predicted label is -1. Thus, $f(\mathbf{x}) = \text{sign}(\mathcal{D}(\mathbf{x}) - \theta)$ for some threshold θ.

Because there is less to infer in the matched-pair scenario (a single distribution over $\mathbf{x}$ instead of two), it is plausible that a matched-pair algorithm can more effectively learn a good classifier. Although the problem has been described in terms of underlying distributions, the ultimate aim (for matched-pair learning, as well as for traditional machine learning), is not to infer that underlying distribution directly, but more modestly to learn a discriminating function $f(\mathbf{x})$ that predicts whether the label for a point $\mathbf{x}$ is -1 or $+1$.

Given that $\boldsymbol{\xi}$ is known, it behooves us to exploit that information. It is suggested here that we can use $\boldsymbol{\xi}$ not only to double the size of our training set, but also to provide a structure to the training set which enables more efficient learning.

In the formalism presented here, $\boldsymbol{\xi}$ is assumed to be deterministic and precisely known. In the discussion of future work in Section 5, an approach is sketched out for extending this formalism to account for uncertainty or stochasticity in $\boldsymbol{\xi}$. Also, the experiments with the gas-phase plume in Section 4.1.4 employ a $\boldsymbol{\xi}$ that varies over different parts of the plume.

3 Data Sets

The experiments in this paper use three different data sets, which are described in Sections 3.1, 3.2, and 3.3. Two of these data sets (in Sections 3.1 and 3.2) are fully simulated, while the application in Section 3.3 uses real hyperspectral data with simulated chemical plumes. In Section 3.4, variants of these data sets are described that use $\boldsymbol{\xi}(\mathbf{x})$ in different ways to generate matched-pair data sets.

3.1 Multivariate Gaussian data with common covariance

A simple but revealing data set is provided by a d-dimensional multivariate Gaussian distribution with an additive treatment. Let $p(\mathbf{x}, -1)$ be normal with mean 0 and $d \times d$ covariance R, and let $\boldsymbol{\xi}(\mathbf{x}) = \mathbf{x} + \mathbf{t}$ for some $\mathbf{t} \in \mathbb{R}^d$. Then it follows from (1) that $p(\mathbf{x}, +1) = p(\mathbf{x} - \mathbf{t}, -1)$, and that $p(\mathbf{x}, -1)$ and $p(\mathbf{x}, +1)$ are both Gaussian with common covariance matrix (R) but different centroids (0 and $\mathbf{t}$, respectively). Thus, the optimal boundary between the two classes is linear. Indeed, the likelihood ratio leads to the optimal detector $\mathcal{D}(\mathbf{x}) = \mathbf{t}^T R^{-1} \mathbf{x}$.

For our simulations, we will employ a diagonal covariance matrix R with geometrically decreasing eigenvalues; that is,

$$R = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & \lambda & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \lambda^{d-1} \end{bmatrix}. \tag{2}$$

In particular, we take a relatively high dimensional $d = 25$ data set with a modest dynamic range of 100; thus $1/\lambda^{24} = 100$. We take $\mathbf{t}$ in the direction $[1, 1, \dots, 1]^T$ (so that it has

[1] This is the standard change-of-variables formula for probability distributions. In (1), $\boldsymbol{\xi}$ is treated as if it were invertible, and for the examples here, that is the case. But the extension to non-invertible $\boldsymbol{\xi}$ presents no serious difficulties. Basically, one would have a sum over all the pre-images.

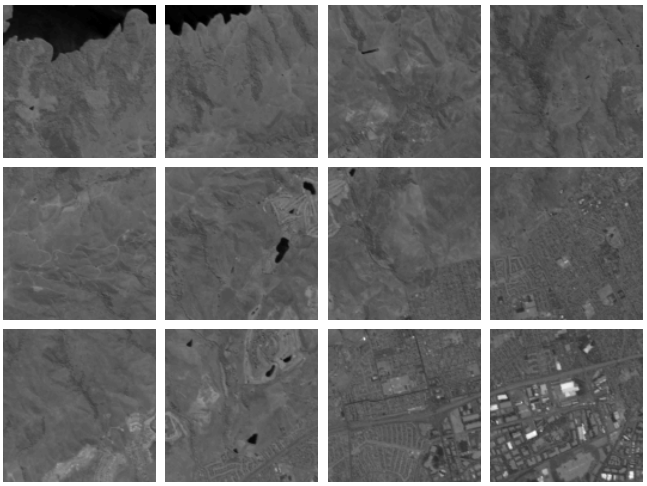

Figure 1: Twelve 128×128 pixel tiles from a 224-channel hyperspectral image collected over Moffett Field in California (AVIRIS flightline `f970620t01p02_r03_sc01`).

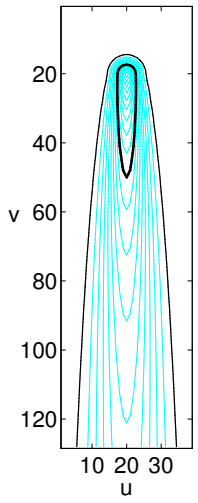

Figure 2: Contour plot shows the spatial variation of plume strength, relative the the maximum value, which here is at position (20,20). Shown are contours of values from 0.05 to 1.0 in steps of 0.05, with a darker line at 0.05 (outside of which is off-plume area for which $T(u,v) < 0.05$), and a dark and thicker solid contour at 0.5 (inside of which is the on-plume area for which $T(u,v) > 0.5$).

nonzero components with all of the eigenvectors of R), but with varying magnitude, so that we can investigate how performance varies with class separation.

3.2 Low-dimensional non-Gaussian data

The previous data set was high dimensional with a linear boundary between classes. We will also consider a low-dimensional data set whose optimal boundary is nonlinear. We begin by drawing $\mathbf{w}$ from a multivariate Gaussian (described above) using $d = 2$ and $1/\lambda = 10$; thus, the variance in the first (long) dimension is 1, and the variance in the second (thin) direction is 1/10. We obtain $\mathbf{x}$ from $\mathbf{w}$ by applying a nonlinear transform to the second coordinate:

$$x_1 = w_1; \quad (3)$$
$$x_2 = w_2 + \sin(w_1). \quad (4)$$

Again, we use $\boldsymbol{\xi}(\mathbf{x}) = \mathbf{x} + \mathbf{t}$, and here we fix $\mathbf{t} = [0, 0.3]^T$. A sample of data is plotted later in Figure 3(a), which shows the optimal boundary between these two classes is nonlinear.

3.3 Chemical plumes in hyperspectral data

The application of matched-pair machine learning to the plume detection problem is illustrated on some real hyperspectral data from the AVIRIS sensor (Vane et al., 1993). This 224-channel data, with a spectral wavelength range from from 390 nm (blue) to 2500 nm (near infrared), was made available on the AVIRIS free standard data products website (`http://aviris.jpl.nasa.gov/html/aviris.freedata.html`). The full image is divided into a dozen 128×128 tiles, shown in Figure 1, and is included in the Supplementary Materials online. Thus the sample size is $n = 16384$ pixels per tile, and the input dimension is $d = 224$.

From these real data sets, we also construct image tiles that include a plume of NO_2 gas. The spatial variation of the plume strength is shown in Figure 3.3. Relative plume strength at pixel position (u, v) is given by the expression

$$T(u,v) = \sqrt{\frac{\eta}{\eta + [v - v_o]_+}} \exp\left(-\frac{(u-u_o)^2 + [v-v_o]_-^2}{\eta + [v-v_o]_+}\right) \quad (5)$$

where $[w]_+ = \max(w, 0)$ and $[w]_- = \min(w, 0)$. The position (u_o, v_o) is where the plume takes its strongest value: $T(u_o, v_o) = 1$. For a plume of specified strength ϵ_o, placed in the scene at position (u_o, v_o), the strength at position (u, v) is given by $\epsilon = \epsilon_o T(u, v)$. In the experiments reported here, we use a characteristic plume width of $\eta = 10$ pixels.

The effect of the plume is to suppress the radiance by an amount proportional to the gas absorption cross section $\mathbf{b}$ in units of cm^2/molecule (this quantity is specific to the chemical species, and is known), and the gas column density ϵ in parts-per-million-meters (ppmm). Specifically,

$$\boldsymbol{\xi}(\mathbf{x}) = \mathbf{x} \cdot \exp(-c\epsilon\mathbf{b}) \quad (6)$$
$$= \mathbf{x} \cdot \exp(-c\epsilon_o T(u,v)\mathbf{b}) \quad (7)$$

where $\mathbf{x}$ is the spectrum of radiance versus wavelength at a given pixel (u, v) in the absence of plume, the '$\cdot$' symbol corresponds to element-wise multiplication of vectors, and the constant $c = 5.64 \times 10^{15}$ cm^{-2} m^{-1}. The gas column density (plume strength) ϵ has spatial variation given by $\epsilon_o T(u, v)$ with $T(u, v)$ defined in (5) for testing, but we will use $T(u, v) = 1$ for matched-pair training. It bears emphasizing that the vector $\mathbf{x}$ is of dimension $d = 224$; there is one component for each wavelength at which intensity is measured. In particular, the (u, v) position of a pixel is *not* encoded in the vector $\mathbf{x}$.

The simulation here is relatively simplistic, and does not take into account, for instance, atmospheric scattering, sensor saturation, or emissive effects. But even very complicated simulations, as long as they can ultimately be expressed in terms of a treatment function $\boldsymbol{\xi}(\mathbf{x})$, can take advantage of the matched-pair approach.

3.4 Matched pairs of training samples

For each of the data sets described in the previous subsections, we can create matched-pair data sets so that the training samples come in pairs of the form $(\mathbf{x}, -1)$ and $(\boldsymbol{\xi}(\mathbf{x}), +1)$. This produces a structure in the training data that is not normally present, and which leads to better classifiers. In general, we will consider these variants:

(A) an "initial" data set with m labeled samples taken from the underlying distribution; some with $y = -1$ and some (possibly much fewer) with $y = 1$. There is no matched-pair structure in this data set.

(B) an "augmented" data set with $2m$ labeled samples, in which the initial m samples are extended by m new points, sampled from the parent distribution, again without any matched-pair structure. Thus, this is the same as (A) but with twice as many samples.

(C) a "matched-pair" data set with $2m$ samples, in which the initial m samples have been extended by m more samples, which are the corresponding opposites of the initial m samples: that is, $(\mathbf{x}, y)$ is matched with $(\boldsymbol{\xi}^{-y}(\mathbf{x}), -y)$.

(D) a "scrambled matched-pair" data set with $2m$ samples that has the same data as the matched-pair data set (D), but the data order has been randomized.

(E) an initially unlabeled or "contaminated" matched-pair data set with $2m$ labeled samples that are created from m unlabeled samples, as described below.

(F) a "transductive" matched-pair data set, which is identical to the data in (E), but is evaluated on the original unlabeled data, instead of a separate out-of-sample data set.

The third and fourth data sets – (C) and (D) – are identical except for data order. For many algorithms, they will produce the same result, but for some algorithms, such as the "online" algorithms described in Section 4.2, the order does matter. Whereas the data in set (C) has an explicit structure to it, one might say that for the data in set (D), that structure exists but is not made available for an algorithm to exploit. Of course, there is no practical advantage to scrambling the data in an operational scenario; the idea is to disentangle the implicit benefits of matched-pair structure in the data from the effects of explicit exploitation of that structure by some of the algorithms.

The contaminated (E) and transductive (F) data sets illustrate the concept that knowing the treatment $\boldsymbol{\xi}$ allows one to do classification of unlabeled data. From an unlabeled data sample $\mathbf{x}$, for instance, one can produce the training pair $\left\{(\boldsymbol{\xi}^{-1}(\mathbf{x}), -1), (\boldsymbol{\xi}(\mathbf{x}), +1)\right\}$. One then learns $f(\mathbf{x})$ from this paired training data, and applies that $f(\mathbf{x})$ either to out-of-sample data points (E), or back to the original unlabeled samples (F). As we will see in Section 4.1.2, the application of $f(\mathbf{x})$ back to the original unlabeled samples is particularly useful for detecting weak signals on cluttered backgrounds. For the detection problem, however, since the default label is -1, we use training pairs given by $\{(\mathbf{x}, -1), (\boldsymbol{\xi}(\mathbf{x}), +1)\}$.

To illustrate these variants, we have plotted examples of the nonlinear data described in Section 3.2. Figure 3(a), which corresponds to item (A) in Section 3.2, shows an initial data set with $m = 10$ labeled samples. In Figure 3(b), corresponding to item (B), the initial $m = 10$ data samples are augmented with another 10 samples – fresh data points, sampled from the parent distribution. The matched-pair data in Figure 3(c), corresponding to item (C), has the same number of data points as Figure 3(b), but they are derived directly from the original data in Figure 3(a) and do not rely on information about the underlying distribution. If the scrambled matched-pair data described in item (D) were plotted, it would be identical to Figure 3(c). If the initial data were unlabeled, as shown in Figure 3(d), then we could still create matched-pair data, by applying both $\boldsymbol{\xi}$ and $\boldsymbol{\xi}^{-1}$ to the unlabeled data; this case corresponds to items (E) and (F) in Section 3.2.

4 Learning with matched-pair data

We will consider both batch and online algorithms. For the batch algorithms, all of the data is provided at once, and order doesn't matter. For the online algorithms, data samples are presented sequentially.

4.1 Batch algorithms

This section will consider learning algorithms that use all of the training data in one batch. Section 4.1.1 will open with some general remarks about the Fisher discriminant, and will develop a simple model for estimated error of the Fisher discriminant when applied to ordinary and matched-pair data. Section 4.1.2 will apply the linear Fisher discriminant to simulated Gaussian data described in Section 3.1 for which the optimal boundary between classes is linear. In Section 4.1.3, we will apply a k-nearest-neighbor (kNN), and a support vector machine (SVM) as implemented in the libSVM package (Chang and Lin, 2001), to the data in Section 3.2, for which the optimal boundary is nonlinear. Then in Section 4.1.4, we will apply the Fisher discriminant to the plume

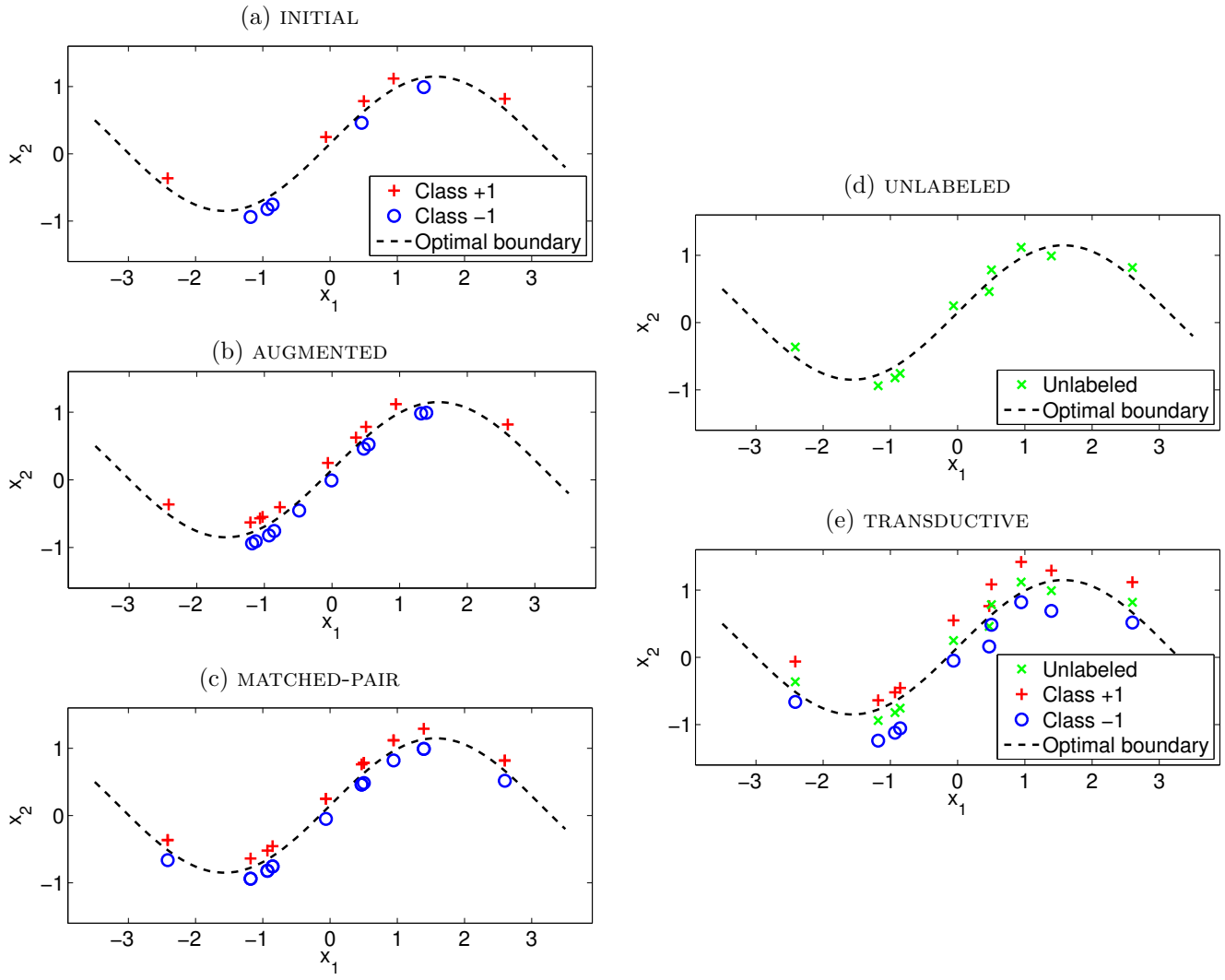

Figure 3: Illustrates different ways, described in Section 3.4, that matched-pair data can be derived from an initial set of observations, using the data described in Section 3.2. **(a)** $m = 10$ unmatched observations; **(b)** the initial $m = 10$ observations are augmented by adding 10 more independent sampled observations; **(c)** the initial $m = 10$ observations are plotted with their corresponding matched pairs. **(d)** the initial $m = 10$ observations are plotted without labels. **(e)** each unlabeled observation in (d) is associated with a matched pair of labeled observations.

detection data described in Section 3.3. For all of these experiments, standard learning algorithms will be used; they are not modified to take explicit advantage of the matched-pair structure in the data, but are observed to achieve improved performance when the data set is composed of matched pairs.

4.1.1 Fisher discriminant

The Fisher discriminant is a simple linear classifier, but its simplicity will help elucidate the effects of matched-pair data on learning.

If $p(\mathbf{x})$ is Gaussian and $\boldsymbol{\xi}(\mathbf{x}) = \mathbf{x} + \mathbf{t}$ (as it is, for instance, with the data in Section 3.1), then the optimal classification is given by a linear filter given by $\mathbf{q} = R^{-1}\mathbf{t}$, where R is the covariance matrix of the Gaussian distribution and the classifier is $f(\mathbf{x}) = \text{sign}(\mathbf{q}^T\mathbf{x} - \theta)$ for some scalar offset θ which depends on the relative importance of false alarms and missed detections.

In the traditional formulation, one does not know $\mathbf{t}$ but is given a set of training examples: $\{(\mathbf{x}_i, y_i); i = 1, \ldots, m\}$. The Fisher discriminant algorithm computes estimates for both the pooled covariance R and for the separation $\mathbf{t}$. In particular, we estimate centroids

$$\boldsymbol{\mu}_{+1} = \frac{\sum_{i=1}^m \mathbf{x}_i \mathcal{I}(y_i = 1)}{\sum_{i=1}^m \mathcal{I}(y_i = 1)} \quad (8)$$

$$\boldsymbol{\mu}_{-1} = \frac{\sum_{i=1}^m \mathbf{x}_i \mathcal{I}(y_i = -1)}{\sum_{i=1}^m \mathcal{I}(y_i = -1)} \quad (9)$$

(where $\mathcal{I}$ is the indicator function), and use those to estimate the pooled covariance

$$\hat{R} = \frac{1}{m}\sum_{i=1}^m (\mathbf{x}_i - \boldsymbol{\mu}_{y_i})(\mathbf{x}_i - \boldsymbol{\mu}_{y_i})^T \quad (10)$$

and separation

$$\hat{\mathbf{t}} = \boldsymbol{\mu}_{+1} - \boldsymbol{\mu}_{-1}. \quad (11)$$

The Fisher discriminant uses $\hat{\mathbf{q}} = \widehat{R^{-1}}\hat{\mathbf{t}}$, which allows two potential sources of error: one in the estimation of the inverse covariance matrix,[2] and one in the estimation of the difference of centroids. Let us write

$$\begin{aligned} \hat{\mathbf{q}} &= \widehat{R^{-1}}\hat{\mathbf{t}} \\ &= (R^{-1} + \Delta R^{-1})(\mathbf{t} + \Delta\mathbf{t}) \\ &= R^{-1}\mathbf{t} + (\Delta R^{-1})\mathbf{t} + R^{-1}\Delta\mathbf{t} + (\Delta R^{-1})\Delta\mathbf{t}. \end{aligned} \quad (12)$$

Each of ΔR^{-1} and $\Delta\mathbf{t}$ vary like $O(1/\sqrt{m})$, so their product varies as $O(1/m)$ and for large m is not as important in the above expression as the leading terms. In particular,

$$\Delta\mathbf{q} = \hat{\mathbf{q}} - \mathbf{q} = (\Delta R^{-1})\mathbf{t} + R^{-1}\Delta\mathbf{t} + O(1/m). \quad (13)$$

Although $\Delta\mathbf{q}$ provides an absolute measure of error in the estimate of $\mathbf{q}$, a more relevant measure is the angular discrepancy between $\hat{\mathbf{q}}$ and $\mathbf{q}$. This is given by $\phi = \cos^{-1}\left(\frac{\mathbf{q}^T\hat{\mathbf{q}}}{|\mathbf{q}|\,|\hat{\mathbf{q}}|}\right)$, but for for small $|\Delta\mathbf{q}| \ll |\mathbf{q}|$, an approximate upper bound is given by

$$\phi \lesssim \frac{\Delta\mathbf{q}}{|\mathbf{q}|} \approx \frac{(\Delta R^{-1})\mathbf{t}}{|R^{-1}\mathbf{t}|} + \frac{R^{-1}\Delta\mathbf{t}}{|R^{-1}\mathbf{t}|}, \quad (14)$$

When $\mathbf{t}$ is not known *a priori*, then the error in $\mathbf{t}$ arises from error in the estimate of the centroids; that is, from (11),

$$\Delta\mathbf{t} = \Delta\boldsymbol{\mu}_+ - \Delta\boldsymbol{\mu}_-, \quad (15)$$

which is independent of the magnitude of $\mathbf{t}$. Thus $\Delta\mathbf{t}/|\mathbf{t}|$ scales like $1/|\mathbf{t}|$. The error in the covariance matrix is also independent of $\mathbf{t}$ and so we have that the term $(\Delta R^{-1})\mathbf{t}/|R^{-1}\mathbf{t}|$ scales independently of $|\mathbf{t}|$. Thus, for small $\Delta\mathbf{q}$, we expect

$$\frac{\Delta\mathbf{q}}{|\mathbf{q}|} \approx \frac{1}{\sqrt{m}}\left(C_1 + \frac{C_2}{|\mathbf{t}|}\right), \quad (16)$$

where C_1 and C_2 characterize how well we can estimate R^{-1} and $\mathbf{t}$, respectively.

When $\mathbf{t}$ *is* known *a priori*, then $C_2 = 0$ in (16). But even when $\mathbf{t}$ is not known, if the training data have the matched-pair structure corresponding to $\{(\mathbf{x}, -1), (\mathbf{x} + \mathbf{t}, +1)\}$, then the estimated $\hat{\mathbf{t}}$ will be exact and, again, we will have $C_2 = 0$.

4.1.2 Fisher discriminant applied to multivariate Gaussian data

The behavior expressed in (16) is seen in the application of Fisher discriminant to the multivariate Gaussian data described in Section 3.1, and this is shown in Figure 4.1.1(a). This figure shows angular discrepancy ϕ between $\hat{\mathbf{q}}$ and $\mathbf{q}$ as a function of class separation, as measured by $\sqrt{\mathbf{t}^T R^{-1}\mathbf{t}}$, the Mahalanobis distance between the centroids. For the matched pair data, we have that $\Delta\mathbf{t} = \mathbf{0}$, and so $C_2 = 0$ and there is no variation in angular error with $|\mathbf{t}|$. For the unmatched data, however, we see that error decreases with $|\mathbf{t}|$ until it reaches an asymptotic value that corresponds to error due to estimation of the inverse covariance R^{-1}. In particular, there is a crossover point at which the $m = 50$ matched pairs of data samples (*i.e.,* $2m = 100$ separate samples) achieves the same performance as $m = 100$ independent samples. For small $|\mathbf{t}|$, the matched pairs are better, but for larger values of $|\mathbf{t}|$, the independent samples provide a better estimate of the covariance, and this leads to better performance in the estimate of $\hat{\mathbf{q}}$. It should still be remarked (and it is also shown in the plot) that $m = 50$ matched pairs are always better than

[2] A common choice for the estimate of the inverse is the inverse of the estimate: that is, $\widehat{R^{-1}} = \hat{R}^{-1}$, but better estimators are sometimes given by shrinkage operators, such as $\widehat{R^{-1}} = ((1-\alpha)\hat{R} + \alpha I)^{-1}$ for appropriate choice of α (Friedman, 1989). But whatever estimator is used, if it is based on m observations, then $O(1/\sqrt{m})$ is the expected scaling of its statistical error.

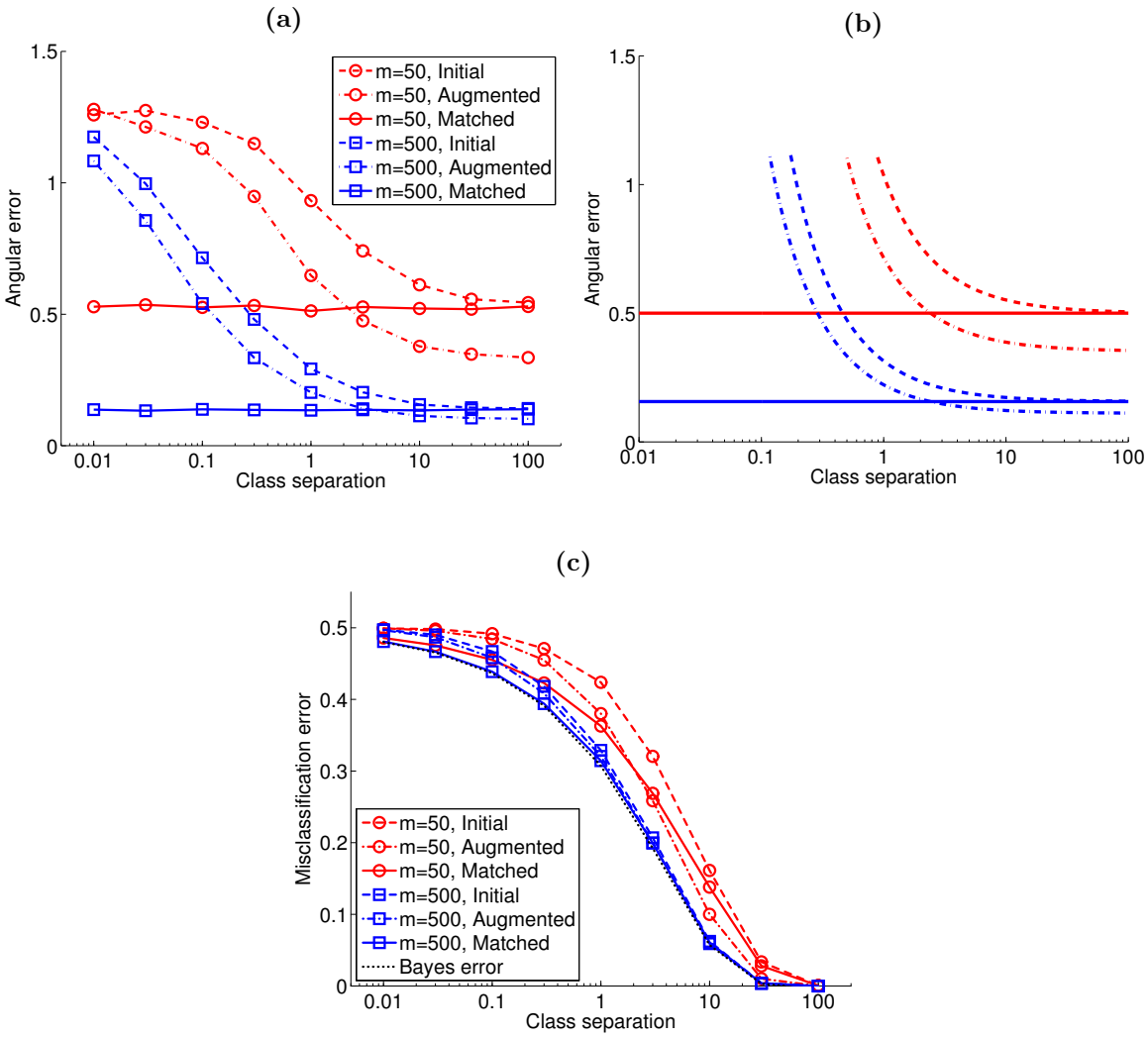

Figure 4: Results of experiments described in Section 4.1.2, using the multivariate Gaussian data described in Section 3.1. **(a)** Angular discrepancy (in radians) between the estimated matched filter $\hat{\mathbf{q}}$ and the actual matched filter is plotted against class separation (given by $\sqrt{\mathbf{t}^T R^{-1} \mathbf{t}}$) for three different sampling methods described in the text. **(b)** The simple model in (16) predicts the main features of how error varies with class separation, for larger values of class separation. **(c)** Misclassification error is plotted for the estimated discriminants.

Table 2: Nonlinear classifiers applied to data described in Section 3.2, using the data augmentation variants described in Section 3.4. Shown is the mean out-of-sample error rate for a k-Nearest Neighbor (kNN) and a support vector machine (SVM) classifier. The Bayes-optimal error for this example is 0.0668 ± 0.0001.

	kNN		SVM	
	$m = 150$ ($k = 3$)	$m = 500$ ($k = 7$)	$m = 150$ ($C = 30$)	$m = 500$ ($C = 30$)
(A) initial	0.1247	0.0898	0.0822	0.0722
(B) augmented	0.1020	0.0820	0.0751	0.0697
(C) matched	0.0984	0.0807	0.0747	0.0696
(E) contaminated	0.1125	0.0852	0.0807	0.0717
(F) transductive	0.1120	0.0850	0.0806	0.0716

$m = 50$ independent points. These trends are echoed in Figure 4.1.1(b), which translates (16) directly into angular error, using $C_1 = C_2 = 3.5$; these values are chosen to correspond to the empirical results in Figure 4.1.1(a). For very small $|\mathbf{t}|$, the there will be no "lever arm" to estimate the direction of $\mathbf{t}$, and the term $R^{-1}\Delta\mathbf{t}/|R^{-1}\mathbf{t}|$ will approach an asymptote whose value depends on R and $\mathbf{t}$; in this large $\Delta\mathbf{q}$ regime, the expression in (16), which suggests that $\Delta\mathbf{q}/|\mathbf{q}|$ should diverge without limit as $|\mathbf{t}| \to 0$, no longer applies.

It bears remarking that the computation of Fisher discriminant, in the ideal case where $\xi(\mathbf{x}) = \mathbf{x} + \mathbf{t}$, does not actually require explicit manufacture of matched-pair data. From pooled covariance R, and known signature $\mathbf{t}$, one can immediately derive the matched filter $\mathbf{q} = R^{-1}\mathbf{t}$.

More generally, what we are seeing here is a trade-off between estimates of the distribution $p(\mathbf{x})$ and estimates of the treatment $\boldsymbol{\xi}$. For traditional machine learning, we need to effectively (if not explicitly) estimate both $p(\mathbf{x})$ and $\boldsymbol{\xi}$; but with matched-pair learning, we already know $\boldsymbol{\xi}$, and only have to estimate $p(\mathbf{x})$. In the example of the Fisher discriminant, the treatment corresponds to a vector $\mathbf{t}$ with d unknown parameters and the distribution corresponds to the covariance matrix R, which has $O(d^2)$ unknown parameters. One might imagine, therefore, particularly for large d, that not knowing the covariance matrix is a much greater a handicap than not knowing the treatment. And therefore, incorporating knowledge of $\boldsymbol{\xi}$ (which is what the matched-pair formalism effectively does) should not be expected to provide much advantage. But the general dictum that machine learning can effectively make inferences about distributions without directly estimating those distributions applies here. We do not ultimately need to estimate R; what we want is an estimate of $\mathbf{q}$, which has only d unknown parameters.

4.1.3 Nonlinear algorithms applied to non-Gaussian data

Using the data described in Section 3.2 and shown in Figure 3, two nonlinear machine learning algorithms were used to investigate the utility of matched-pair learning. The results, shown in Table 2, were based on 10000 trials with each of five kinds of data sets described in Section 3.4: (A) an initial m samples, (B) an augmented data set with m new samples added to the initial samples, (C) matched-pair training set with $2m$ samples derived from the initial samples, (E,F) a contaminated matched-pair training set with $2m$ samples derived from an unlabeled version of the initial samples. Each of the training sets are used to train a k-nearest neighbor (kNN) and a support vector machine (SVM) classifier. The reported error rates are an average over all the trials. For training sets (A), (B), (C), and (E), the error is based on out-of-sample data; for the transductive case (F), performance is based on how well the initially unlabeled data is labeled. Standard errors are provided in the Supplementary Materials, and range from 0.0001 to 0.0003.

One thing that is clear from this table is that the matched-pair training (C) significantly outperformed the training on the initial data (A). In fact, the matched-pair training was slightly (but not substantially) better than the augmented data training (B), but this comparison is not as relevant as the comparison with the initial data, since one does not in general have arbitrary access to new data samples. The matched-pair approach provides an effective augmentation based on the existing samples. The matched-pair training (C) also outperformed the contaminated matched-pair training (E), which is hardly surprising, since the matched-pair training uses information (the labels on the initial training data) that is not used in the training data created from unlabeled initial data. A visual comparison of Figure 3(c) and Figure 3(e) also explains why (C) is expected to outperform (E); the training data is closer to the boundary that one is trying to learn. Comparing the performance of the classifier trained using the contaminated training data in (E,F), we see that the out-of-sample performance (E) is nearly identical to the transductive performance (F), with perhaps a slight advantage to transductive.

The kNN classifier used the parameter k that optimized performance for the initial samples; thus, the improvement that the various augmentation and matched-pair schemes exhibited over the initial is not over-estimated, and may even be slightly under-estimated. The SVM used a Gaussian radial basis function kernel of width $\sqrt{2}$, which is the default in the libSVM package (Chang and Lin, 2001) for data sets in two dimensions; and the regularization constant C was chosen to optimize the performance of the initial samples. In the online Supplementary Materials, a more extensive table is provided with a range of k and C values, which shows that the results are fairly insensitive to choice of k and C. The supplementary table also includes standard error estimates for each number

in the table.

4.1.4 Fisher discriminant applied to hyperspectral data

The matched-pair formalism provides an effective way to learn a signal detector. While signal detection is essentially a traditional classification problem, with the two classes of interest being signal-present and signal-absent, there are some practical differences. One is that signal-absent is usually the default, and there is a plethora of data in the signal-absent class and a corresponding paucity of data in the signal-present class. Another difference is that the signal-present class might actually have signals of different strengths. A further difference, which the matched-pair formalism can productively exploit, is that the effect of signal on background (*i.e.,* the treatment $\boldsymbol{\xi}$) may be known. This enables us to use that treatment to create an abundance of artificial signal-present samples.

In this section, a number of approaches will be used to train an NO_2 plume detector. The first (P) is *plume*-based; it is the traditional approach of using on-plume and off-plume pixels from data where the plume location is known. The second approach (MP) uses *matched-pair* training from data that is known to be plume-free. The third approach (XMP) employs *contaminated* matched pairs; this is similar to MP, but in this case the matched pairs are applied to an image that already includes a plume (at some unknown location).

The performance of the detectors will be tested in three different ways. The in-sample (IS) performance describes how well the detector works on the data with which it was trained; this provides a benchmark for comparison, but does not represent a useful scenario since the training data is already labeled. By contrast, the out-of-sample (OS) performance *is* useful: this is the plume detection performance on imagery that was not used in training. Finally, the nature of the signal detection problem permits a third kind of training/testing combination; the transductive (T) approach begins with unlabeled data, uses knowledge of the treatment $\boldsymbol{\xi}$ to create artificial labeled data, builds a detector trained on that artificial data, and applies the result back to the original unlabeled data.

Relating back to the cases listed in Section 3.4, we have that P-OS corresponds to scenario (A), MP-OS corresponds to scenario (C), XMP-OS corresponds to scenario (E), and XMP-T corresponds to scenario (F).

A combination that is particularly effective for plume detection is transductive training with contaminated matched pairs (XMP-T), and that is illustrated in Figure 5. Panel (a) shows band 8 (at 459 nm, an absorption band for NO_2) of an AVIRIS tile with no plume; this is the upper-left tile in Figure 1. Panel (b) is the same as (a) but with a simulated plume at 20 ppmm (parts-per-million-meters), just barely visible as a darker smudge in the lake (the arrow identifies its

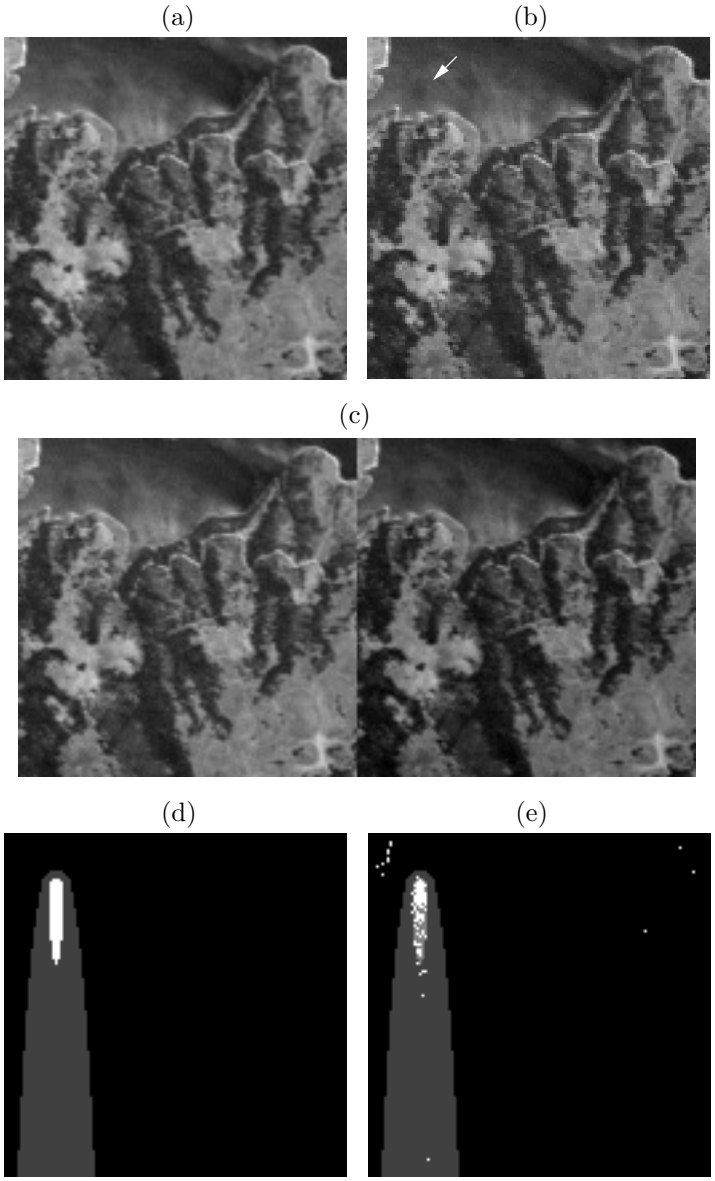

Figure 5: This figure illustrates how contaminated matched-pair transductive (XMP-T) learning works on the hyperspectral data described in Section 3.3. See text for further explanation.

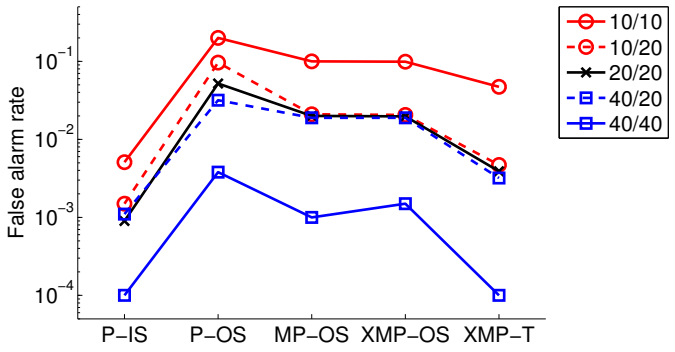

Figure 6: False alarm rate (at the threshold for which detection rate is 0.5) with different strengths of plumes (legend indicates training/testing strength) for various training modes: P is plume, MP is matched-pair, XMP is contaminated matched pair; IS is in-sample, OS is out-of-sample, and T is transductive.

position). In panel (c), a matched-pair image is generated by duplicating the image in (b) and applying the effect of plume to every pixel in the duplicated image. That is, (7) is applied with $T(u, v) = 1$. One can see that the right half of this double image is darker, indicating the absorption effect of the plume in this band.[3] The XMP detector (which is unaware of the existence of a plume in the left half of the image) will attempt to distinguish for a given pixel whether it came from the left or the right half of this double image. Note that although this figure shows only a single band, the detector employs all 224 bands in the AVIRIS image. Panel (d) is the truth map. The plume has spatially varying strength; the white corresponds to strong plume ($T(u, v) > 0.5$), and the black to essentially no plume ($T(u, v) < 0.05$). For evaluation purposes, the white pixels are treated as on-plume, and the black pixels as off-plume. The gray area is intermediate; it does not count in the scoring of plume detection performance. Finally, panel (e) shows results of matched-pair training using the image in (c) for training, and applying the result back to (b). The white pixels are the detections. The twelve white pixels that appear in the black area are false alarms. The white pixels that appear in the lightest gray area are correct detections. There are also a number of white pixels observed in the intermediate gray area; these are not counted either as detections or as false alarms. Although we need the truth map to *evaluate* the performance of our detector, we do not use knowledge of the plume location to *train* the XMP detector.

[3]This protocol is analogous to the transductive training illustrated in Figure 3(e), but with an important difference. In Figure 3(e), positive and negative samples were equally likely, and the matched pairs were given by $\{(\boldsymbol{\xi}^{-1}(\mathbf{x}), -1), (\boldsymbol{\xi}(\mathbf{x}), +1)\}$. For the plume detection problem, most samples are negative, and the matched pairs we use are $\{(\mathbf{x}, -1), (\boldsymbol{\xi}(\mathbf{x}), +1)\}$.

For all three training approaches, we use twelve-fold cross-validation, with each fold corresponding to a different training tile. We train on a single 128×128 tile, and apply both to that tile (for in-sample and transductive) and to all the other tiles (for out of sample). The experiments are performed five times, each time with the plume in a different location ($u_o = 20, 40, 60, 80, 100; v_o = 20$), and the results from these five runs are averaged. We consider three different plume strengths ϵ_o, ranging from barely to easily detectable (10, 20, and 40 ppmm). The class separability associated with these plume strengths can be visualized with histograms of matched filter output for both off-plume and on-plume pixels; these histograms are provided in the Supplementary Materials.

Each plume includes 136 on-plume pixels, corresponding to $T(u, v) > 0.5$ (see Figure 3.3). Each detector's threshold θ is adjusted to a level that produces a detection rate of one half (*i.e.,* of 68 pixels), and the fraction of pixels that are (falsely) detected in the off-plume area (*i.e.,* black in Figure 5(d)) provides the error measure that is reported in Figure 6. Although this provides a consistent way to compare performance, it bears remarking that the value of this threshold would not be known in operational practice.

A number of trends are evident in Figure 6. As a sanity check, we see that as the strength of the plume increases, it is more easily detectable. At 10 ppmm, the plume is essentially lost in the background, with a false alarm rate that is very high. (Note that even a five percent false alarm rate would be unacceptably large in most scenarios because there are so many off-plume pixels in the image.) For the stronger 40 ppmm plume, by contrast, the false alarm rate drops by roughly two orders of magnitude. Although training and testing are generally performed with the same plume strength, we included two experiments where we trained with either weaker (10 ppmm) or stronger (40 ppmm) plumes than we used for testing (20 ppmm). The main observation is that the training and testing strengths do not need to be precisely matched, although we observe in all but the in-sample case that training with a stronger plume led to a detector that performed better.

The lowest error is achieved for in-sample plume training (P-IS); this is no surprise, since the training and testing sets are identical. This is also of no operational use because it requires one to know where the plume is before attempting to detect the plume. It does, however, provide a lower bound on the expected error for the given plume strength.

One clear result is that MP-OS outperforms P-OS; that is, matched-pair training is more effective than training based on plumes. It bears remarking that the matched-pair training employed a single plume strength over the whole image, whereas the testing employed a model plume whose strength varied spatially over the extent of the plume.

One issue with MP training is that it assumes that there is no plume present in the image where training occurs. This is operationally plausible, but it is also possible that a plume

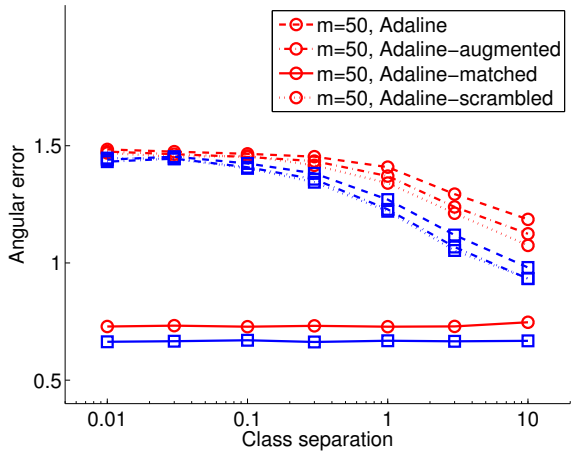

Figure 7: Angular error versus class separation for the online Adaline classifier, applied to the data described in Section 3.1, using the four matched-pair scenarios described in Section 3.4. As in Figure 4.1.1, the tests are applied to data sets with $m = 50$ ($\circ$) samples, and to data sets with $m = 500$ ($\square$). Plotted results are based on out-of-sample error, averaged over 200 trials.

may be present in an area that is assumed to be plume-free. Comparing MP-OS with XMP-OS, we see that the effect of the contamination is negligible, except at the largest plume strength.

Using matched-pair training on data that may already contain a plume (of unknown location and strength) enables it to be used with the same image that it was trained on. Thus, XMP-T is an operationally viable protocol. And, for the experiments in this section, it also achieved substantial improvement over the out-of-sample protocols. By contrast, the transductive advantage in Table 2 was negligible.

4.2 Online learning algorithms

For some problems, the data arrives one or a few samples at a time, and one seeks a classifier that can be trained incrementally. An example involving real-time target detection in hyperspectral imagery is provided by Schaum (2006). Traditional learning algorithms can in principle be used in this scenario, by simply re-training from scratch with an ever-growing training set, but this also imposes ever-growing demands on memory and computing resources, and so online algorithms have been developed for efficient learning in this situation. For these algorithms, the order of data presentation can really matter, and as we will see, matched-pair approaches can improve their performance.

4.2.1 Adaline

The ADAptive LINear Element (ADALINE) algorithm of Widrow and Hoff (1960) is designed to optimize a loss function that corresponds to the Fisher discriminant. So for a problem in which the two classes are both Gaussian with different means but the same covariance, it is an appropriate choice for investigating the utility of matched pairs. Initializing the iteration at $\mathbf{q}_0 = \mathbf{0}$, the matched filter estimate is updated using

$$\mathbf{q}_n = \mathbf{q}_{n-1} + \gamma_n \left(y_n - \mathbf{q}_{n-1}^T \mathbf{x}_n\right) \mathbf{x}_n \tag{17}$$

where $(\mathbf{x}_n, y_n)$ is the n'th data sample, and γ_n is a time-dependent gradient multiplier. Following the usual prescription for stochastic approximation (Robbins and Monro, 1951; Bottou and Le Cun, 2004), we take $\gamma_n = \gamma_o/n$, and for the experiments reported here, we adjusted γ_o by hand. The algorithm is a stochastic gradient descent to minimize the average $\left\langle (y - \mathbf{q}^T\mathbf{x})^2 \right\rangle$.

If we take the expected value of (17), we obtain

$$\begin{aligned} \langle \mathbf{q}_n \rangle &= \langle \mathbf{q}_{n-1} \rangle + \gamma_n \left\langle \left(y_n - \mathbf{q}_{n-1}^T \mathbf{x}_n\right) \mathbf{x}_n \right\rangle \\ &= \langle \mathbf{q}_{n-1} \rangle + \gamma_n \left(\langle y_n \mathbf{x}_n \rangle - \left\langle \mathbf{x}_n \mathbf{x}_n^T \mathbf{q}_{n-1} \right\rangle \right) \\ &= \langle \mathbf{q}_{n-1} \rangle + \gamma_n \left(\langle y_n \mathbf{x}_n \rangle - \left\langle \mathbf{x}_n \mathbf{x}_n^T \right\rangle \langle \mathbf{q}_{n-1} \rangle \right) \end{aligned} \tag{18}$$

where the last line follows from the independence of $\mathbf{x}_n$ and $\mathbf{q}_{n-1}$. The convergence is to $\mathbf{q} = \left\langle \mathbf{x}\mathbf{x}^T \right\rangle^{-1} \langle y\mathbf{x} \rangle$ even as neither $\left\langle \mathbf{x}\mathbf{x}^T \right\rangle$ nor $\langle y\mathbf{x} \rangle$ are directly estimated. This Fisher discriminant solution is the optimal detector for Gaussian data.

As seen in Figure 7, the algorithm performs best (by a substantial amount) when using matched-pair data. But the order of data presentation is important; for the "Adaline-matched" curve, each data point in the original set was immediately followed by its matched-pair counterpoint. The experiments shown in Figure 7 also included a scrambled set of matched-pair data – the same data but in a random order – and that also performed poorly compared to the ordered matched-pair result.

4.2.2 Adaline-Dipole

Since we know that the data are matched pairs, we can build this structure directly into the algorithm instead of storing it (redundantly) in the input data set. Given a data set and a process $\boldsymbol{\xi}$, we can rewrite the update rule in (17) as:

$$\begin{aligned} \mathbf{q}_n = \mathbf{q}_{n-1} &+ \frac{1}{2}\gamma_n \left(y_n - \mathbf{q}_{n-1}^T \mathbf{x}_n\right) \mathbf{x}_n \\ &+ \frac{1}{2}\gamma_n \left(-y_n - \mathbf{q}_{n-1}^T \boldsymbol{\xi}^{-y_n}(\mathbf{x}_n)\right) \boldsymbol{\xi}^{-y_n}(\mathbf{x}_n). \end{aligned} \tag{19}$$

In the case where we have our matched pairs given by $(\mathbf{x}, -1)$ and $(\mathbf{x} + \mathbf{t}, +1)$, then we can write the update rule

$$\mathbf{q}_n = \mathbf{q}_{n-1} + \gamma_n \left(\frac{1}{2}\mathbf{t} - (\mathbf{q}_{n-1}^T \mathbf{x}_n - \mathbf{t}\mathbf{t}^T)\mathbf{x}_n \right) \tag{20}$$

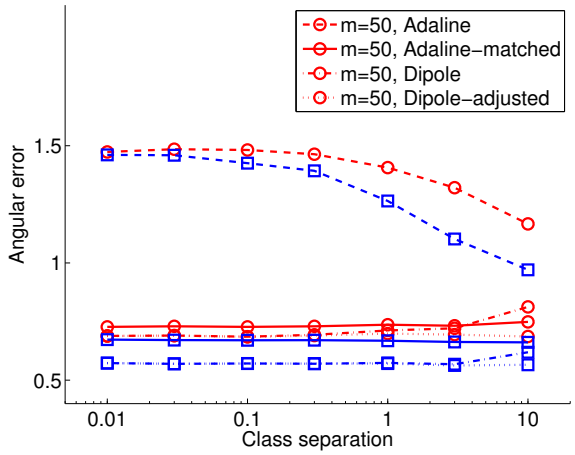

Figure 8: Performance of variants of the Adaline-dipole algorithm, applied to the data described in Section 3.1. The dashed and solid lines correspond to performance of the traditional Adaline algorithm using initial samples and using the matched-pair samples, respectively – these curves also appear in Figure 7. Also, as in Figure 7, the tests are applied data sets with to $m = 50$ (○), and to data sets with $m = 500$ (□).

We call this the Adaline-dipole algorithm. This is similar to the straight Adaline update rule in (17); the main difference is that $y_n\mathbf{x}_n$ is replaced by its expected value: $\frac{1}{2}\mathbf{t}$. (Thus, rather than "learn" the expected value of $y_n\mathbf{x}_n$ from multiple iterations, we take advantage of our knowledge of $\boldsymbol{\xi}$, and use that expected value directly.) The extra $\mathbf{t}\mathbf{t}^T$ term has little effect; it alters the magnitude of $\mathbf{q} = \lim_{n\to\infty}\mathbf{q}_n$ but not its direction.

The performance of two variants of Adaline-dipole is shown in Figure 8. The dot-dashed line is the simple Adaline-dipole algorithm applied to the initial data, literally ignoring the labels on that data. The dotted line is a "dipole adjusted" variant of Adaline-dipole that is applied to the initial data, but uses either a given data point or its matched-pair counterpoint, according to which one had a label of -1. With this small adjustment, which still only needs half as many updates as the Adaline algorithm with matched pairs, good performance is seen across the entire range of class separation.

5 Conclusions and future work

In analogy to the distinction that is made in statistical hypothesis testing between matched-pair and independent-sample statistics, we have investigated the use of matched pairs for machine learning. We have found that by taking advantage of known structure in some problems, matched pairs can be generated which not only double the effective size of the training set but which enable the learning algorithm to more effectively find a good classifier. For batch algorithms, just supplying appropriately paired data can improve performance; for online algorithms, it is important that the pairs be explicitly matched. In some cases, the learning algorithm can be modified so that this efficient learning can be applied directly to the initial (smaller) data set. The Adaline-dipole is an example of an initial algorithm that achieves this result.

Although the motivation for this approach is a specific problem in remote sensing, we believe that the potential applications are broader than that. As well as for other signal detection problems (where the target is well characterized, but the background is not), the approach can be used whenever the aim is to determine whether a "treatment" has been applied in cases where the effect of that treatment is known. Has a photograph been digitally sharpened? or jpeg compressed? Has an audio signal been band-pass filtered? Is a handwritten character upside down?

If there are multiple treatments whose individual effects are known (*e.g.,* different white-balance corrections applied to a picture), then one can use matched triplets (or quadruplets, *etc.*) in a multi-class learning context. The extension to regression may also be straightforward; here, the treatment is a variable amount and the goal is to infer that amount.

Another approach for exploiting known structure in data is to to design a kernel that takes this structure into account; *e.g.,* see Schölkopf and Smola (2002, chap. 11). We speculate that dipole kernels might provide an an efficient implementation for a kernel-based algorithm (such as a support vector machine) that directly exploits the match-pair structure in the data.

In the presentation here, $\boldsymbol{\xi}(\mathbf{x})$ is assumed to be both deterministic and precisely known; for real problems, this is at best an approximation. But a stochastic model for $\boldsymbol{\xi}(\mathbf{x})$, while not investigated here, should be relatively straightforward to incorporate into the algorithms. For the matched pair $\{(\mathbf{x}, -1), (\mathbf{x}', +1)\}$, instead of $\mathbf{x}' = \boldsymbol{\xi}(\mathbf{x})$, one would draw $\mathbf{x}'$ from a distribution $p(\mathbf{x}'|\mathbf{x})$. Depending on the variability in this distribution, multiple samples might profitably be drawn.

Informally, one often thinks of machine learning as a way to infer the "rule" that distinguishes two cases, based solely on representative examples of those cases. In the matched-pair formalism, what might be considered the rule, namely the treatment function $\boldsymbol{\xi}(\mathbf{x})$ that maps members of one class into another, is already known. The problem remains nontrivial, however, because the probability distribution $p(\mathbf{x})$, from which the data are presumed to be drawn, is not known. What machine learning provides is a principled approach for making important inferences about an unknown distribution without directly estimating that distribution. What the matched-pair formalism adds to this is a systematic way to incorporate domain knowledge into the inference process.

Supplementary Materials

The following supplementary materials are available online. The data sets are contained in a single zip file, and the supplementary tables and figures are included in a single pdf file.

Hyperspectral images: Twelve AVIRIS images shown in Figure 1. (Matlab 'mat' files, included in zip file)

Plume: Data used to simulate the plume; this includes the absorption spectrum for NO_2 (the vector $\mathbf{b}$ that appears in (7)), the spatial map of plume strength given by the function $T(u, v)$ given in (5) and shown in Figure 3.3, the plume mask seen in Figure 5(d), and the three-dimensional product $cT(u, v)\mathbf{b}$ that appears in (7). (Matlab 'mat' file, included in zip file)

Additional results: A pdf file containing: i) a description of the data in the zip file; ii) an extended Table 2 with error bars and a range of k and C parameter values; iii) an alternate Figure 4.1.1(c) with logarithmic vertical axis; iv) density plots of matched-filter output of on-plume and off-plume pixels described in Section 4.1.4; and v) a table of values shown in Figure 6, including error bars.

Acknowledgments

I am grateful to Bernard Foy for many valuable conversations about chemical plumes in hyperspectral imagery, and to Don Hush and Reid Porter for insightful discussions on transductive learning. I am very pleased to acknowledge the reviewers and editors of Technometrics for their careful reading of this manuscript, and for their numerous and thoughtful suggestions; thanks to them, this is a much better paper. This work was supported by the United States Department of Energy, through the Los Alamos Laboratory Directed Research and Development (LDRD) program.

References

Beer, A. (1852). Bestimmung der absorption des rothen lichts in farbigen flussigketiten. *Ann. Physik*, 86:78–88.

Bottou, L. and Le Cun, Y. (2004). Large scale online learning. In Thrun, S., Saul, L., and Schölkopf, B., editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA.

Box, G. E. P., Hunter, W. G., and Hunter, J. S. (1978). *Statistics for Experimenters*. Wiley.

Chang, C.-C. and Lin, C.-J. (2001). *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Crawley, M. J. (2005). *Statistics: an introduction using R*. Wiley, West Sussex, UK.

Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern Classification*. John Wiley & Sons, New York.

Foy, B. R., Theiler, J., and Fraser, A. M. (2009). Decision boundaries in two dimensions for target detection in hyperspectral imagery. *Optics Express*, 17:17391–17411.

Friedman, J. H. (1989). Regularized discriminant analysis. *J. American Statistical Association*, 84:165–175.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, New York.

Hayden, A., Niple, E., and Boyce, B. (1996). Determination of trace-gas amounts in plumes by the use of orthogonal digital filtering of thermal-emission spectra. *Applied Optics*, 35:2802–2809.

Manolakis, D. (2008). Signal processing algorithms for hyperspectral remote sensing of chemical plumes. *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1857–1860.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *Ann. Math. Stat.*, 22:400–407.

Schaum, A. (2006). A remedy for nonstationarity in background transition regions for real time hyperspectral detection. *IEEE Aerospace Conference*. doi:10.1109/AERO.2006.1655929.

Schölkopf, B. and Smola, A. J. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA.

Vane, G., Green, R. O., Chrien, T. G., Enmark, H. T., Hansen, E. G., and Porter, W. M. (1993). The Airborne Visible/Infrared Imaging Spectrometer (AVIRIS). *Remote Sensing of the Environment*, 44:127–143.

Vapnik, V. (1999). *The Nature of Statistical Learning Theory*. Springer, New York, 2nd edition.

Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S*. Springer, 4th edition.

Widrow, B. and Hoff, M. E. (1960). Adaptive switching circuits. In *1960 IRE WESCON Convention Record*, pages 96–104, New York. IRE.