Modern Signal Processing
MSRI Publications
Volume **46**, 2003

Integrated Sensing and Processing for Statistical Pattern Recognition

CAREY E. PRIEBE, DAVID J. MARCHETTE,
AND DENNIS M. HEALY, JR.

Abstract. This article presents a simple version of Integrated Sensing and Processing (ISP) for statistical pattern recognition wherein the sensor measurements to be taken are adaptively selected based on task-specific metrics. Thus the measurement space in which the pattern recognition task is ultimately addressed integrates adaptive sensor technology with the specific task for which the sensor is employed. This end-to-end optimization of sensor/processor/exploitation subsystems is a theme of the DARPA Defense Sciences Office Applied and Computational Mathematics Program's ISP program. We illustrate the idea with a pedagogical example and application to the HyMap hyperspectral sensor and the Tufts University "artificial nose" chemical sensor.

1. Introduction

An important activity, common to many fields of endeavor, is the act of refining high order *information* (detections of events, classification of objects, identification of activities, etc.) from large volumes of diverse *data* which is increasingly available through modern means of measurement, communication, and processing. This *exploitation* function winnows the available data concerning an object or situation in order to extract useful and actionable information, quite often through the application of techniques from statistical pattern recognition to the data. This may involve activities like detection, identification, and classification which are applied to the raw measured data, or possibly to partially processed information derived from it.

When new data are sought in order to obtain information about a specific situation, it is now increasingly common to have many different measurement degrees of freedom potentially available for the task. Some appreciation of the dimensionality of available data can be obtained by considering measurements

This work is partially supported by DARPA Grant F49620-01-1-0395.

from one sensor, the hyperspectral camera, which is gaining broad application in fields ranging from geological remote sensing to military target identification. This sensor produces an output comprised of hundreds of megapixel images of a scene, each image corresponding to the appearance of that scene in light from a narrow band of frequencies. Taken together, these images present a finely resolved spectrum for each pixel in the scene. The data sets are often presented as cubes and can have on the order of a billion voxels per scene. Of course for real scenes, the billions of degrees of freedom exhibit correlations; nevertheless, the raw data is presented in an overwhelmingly high dimensional space.

This situation is magnified when one considers the diversity of sophisticated sensing mechanisms which might be applied to a given task. For example, remote sensing of terrain may be performed with natural light cameras, infrared cameras, hyperspectral imagers, fully polarimetric imaging radar, or combinations of all of these. This gives us many different views of the scene, but also presents a challenging requirement for effective processing and exploitation algorithms enabling reliable and affordable extraction of information from the high-dimensional spaces of sensed data.

In many situations, constraints on the available time, bandwidth, human and machine resources, and on the prior relevant experience all significantly limit the ability to deal intelligently with the many potential sensing degrees of freedom. This is particularly the case in time-critical applications. In fact, one often finds that not all of the available sensor degrees of freedom are equally useful in a given situation, suggesting the need for a reasoned approach for choosing those particular measurement types to be made and/or communicated and/or processed.

In this paper we show that it is sometimes possible to identify a particularly informative subspace of the space of all possible sensor measurements when it comes to the application of exploitation tasks on the sensed data. We will present examples in which performance is enhanced significantly by finding and working in the corresponding reduced-dimensionality subspace of sensed data. Even more, we will demonstrate in several cases that the determination of this particularly informative subspace then suggests the selection of a further subspace of measurements to improve exploitation performance yet further. This is somewhat analogous to the game of “20 questions,” in which we progressively refine the scope and specificity of our questions based on partial understanding derived from previous attempts to narrow down the possibilities.

This process of focusing and targeting measurements is in fact often realizable in practice, due in part to significant engineering advances made in adaptive “smart” sensor technology. Current and projected capabilities for modifying the way certain important sensors look at the world motivate the development of mathematical methodology for guiding the adaptive selection of the types measurements made by an adaptive sensor/processor subsystem with an eye to enhancing and simplifying the exploitation of the resulting data. We present examples in which the way a sensor views a scene determines the abstract space in which the exploitation is ultimately addressed. In these cases, a judicious choice of sensor viewpoint improves exploitation performance dramatically.

Effective realization of the next generation of sensor/exploitation systems will require balanced integration and joint optimization of adaptive sensor front end functions with the pattern recognition tasks applied to sensor measurements in the system's back end. Development of methodologies for end-to-end joint optimization of sensor/processor/exploitation subsystems with respect to task-specific metrics, is a key theme of the DARPA Applied and Computational Mathematics Program's "Integrated Sensing and Processing" (ISP) effort. Various aspects of this program are currently being pursued by several groups of researchers in academia, industry, and government. Preliminary results suggest that certain applications in target detection and identification may derive significant performance enhancements by applying this concept to take full advantage of adaptive sensor technology.

In this paper, we illustrate one aspect of the ISP idea, in which the exploitation subsystem is concerned with supervised statistical pattern recognition (classification) and the observations take their value in a space with some linear ordering properties, such as multivariate time series. We illustrate the idea with a pedagogical example and application to the HyMap hyperspectral sensor (in which case the functional domain is spectral rather than temporal) and the Tufts University "artificial nose" chemical sensor. Other applications include gene expression analysis via DNA microarrays collected at multiple time instances, functional brain imaging collected at multiple time instances, etc.

2. Statistical Pattern Recognition

Pattern recognition starts with observations and returns class labels. Statistical pattern recognition addresses the problem in a probabilistic framework and applies to it statistical methods. Here we provide a brief description of the basic set up of statistical pattern recognition. For additional details, see, e.g., Fukunaga (1990), Devroye et al. (1996), Duda et al. (2000), Hastie et al. (2001), and references therein.

Let the pair (X, Y) be distributed according to probability distribution F; $(X, Y) \sim F$. Intuitively, X represents measurements made on some phenomenon of interest and Y indicates higher order information about that phenomenon, such as its membership in one of several disjoint classes.

More formally, the *feature vector* X is a Ξ-valued random variable. Usually $\Xi = \mathbb{R}^d$ or some subset thereof. More generally, Ξ may allow for more elaborate data structures such as multivariate time series, images, categorical data, dissimilarity data, etc. We will consider cases in which feature observations are multivariate time series and spectral responses. For categorical data Ξ is simply a set (unordered). In some applications, Ξ may consist of mixed data — some categorical, some continuous and some time series. For example, in a medical application one might have sex (categorical), temperature (continuous), and an EKG (time series).

The *class label* Y is a $\{1,\ldots,J\}$-valued random variable, with $J>1$ usually finite. The label Y indicates the class to which the associated feature vector X belongs. The prior probabilities of class membership are given by $\pi_j := P[Y=j]$. We denote by F_j the *class-conditional distributions* of $X|Y=j$.

We partition statistical pattern recognition into two main categories: *supervised* and *unsupervised*. The distinguishing feature between these two categories is that for supervised pattern recognition training data exist for which the class labels Y are observed, while this is not the case in the unsupervised case. We refer to the supervised case as *classification* and the unsupervised case as *clustering*.

2.1. Classification. In the supervised case, *training data* are available. The training data set is given by $\mathcal{D}_n := \{(X_1,Y_1),\ldots,(X_n,Y_n)\}'$. That is, we have available observations for which the true categorization is known. The goal is to develop a *classifier* g which will take an unlabelled feature vector X, with true but unobserved class label Y, and estimate its class label by $\widehat{Y}=g(X)$. We hope that $\widehat{Y}=Y$ with high probability. Obviously, g should use the available training data and will have functional dependence on the particular observed training data set as well as on the measured features we are trying to classify; thus

$$g : \Xi \times (\Xi \times \{1,\ldots,J\})^n \to \{1,\ldots,J\}.$$

The use of training data to build the classifier is referred to as *training*.

In order for statistical pattern recognition methodologies to have any guarantee of success, we must assume that the training data are *representative*. Usually this means that $(X_i,Y_i) \overset{\text{iid}}{\sim} F$. Alternatively, writing $I\{E\}$ as the indicator function for event E, the *class-conditional sample sizes* given by $N_j(\mathcal{D}_n) := \sum_{i=1}^n I\{Y_i=j\}$ may be *design variables* rather than random variables, in which case the conditional random variables $X_i|Y_i=j$ are independent and identically distributed (iid) according to the class-conditional distributions F_j. In the former case the class-conditional sample sizes $N_j(\mathcal{D}_n)$ yield consistent estimates of the priors—$\widehat{\pi}_j(\mathcal{D}_n) := N_j(\mathcal{D}_n)/n \to \pi_j$ almost surely as $n\to\infty$. In the latter case *a priori* knowledge of the prior probabilities must be assumed.

Given a training data set $\mathcal{D}_n$, the *probability of misclassification* for classifier g is given by

$$L(g|\mathcal{D}_n) := P[g(X;\mathcal{D}_n) \neq Y|\mathcal{D}_n].$$

The *Bayes optimal probability of misclassification* is given by

$$L^\star = \min_{g:\Xi\to\{1,\ldots,J\}} P[g(X)\neq Y];$$

notice that for the purposes of defining this bound, we consider classifiers which are not constrained by a particular training set. A *Bayes rule* is any map $g^\star$ with $L(g^\star) = L^\star$. The Bayes rule can be obtained from the class-conditional distributions F_j and the prior probabilities π_j as

$$g^\star(x) = \arg\max_j \pi_j \, dF_j(x).$$

Notice that $g^\star$ depends on the distribution of (X, Y), but not on the training data set.

The goal of classification, then, is to devise a methodology for taking training data $\mathcal{D}_n$ and constructing a classifier g such that $L(g|\mathcal{D}_n)$ is as close to $L^\star$ as possible. In particular, we desire *consistency*: $L(g; \mathcal{D}_n) \to L^\star$ as $n \to \infty$ (in probability or with probability one).

2.2. The curse of dimensionality. A common misconception in statistical pattern recognition is that "more is better". It is intuitively obvious — and wrong — that if ten features per observation are good then a hundred features are even better. This is a result of one manifestation of the so-called *curse of dimensionality* (Bellman (1961), Scott (1992)).

The curse has several manifestations. Silverman (1986) considers probability density function estimation, and provides a table for the number of observations needed to obtain a point estimate with a given accuracy as the dimension increases. The estimator considered is a nonparametric one, the kernel estimator. It is shown that the number of observations required grows from 4 for univariate data to over 800,000 for ten-dimensional data. Thus, to achieve a given accuracy for a kernel estimator at a single point, the required number of observations grows exponentially in the dimension.

Another consequence of the curse of dimensionality is discussed in Scott (1992), where he points out statistical ramifications of the fact that the volume of a cube in high dimensions resides primarily in the corners, the volume of a sphere resides mostly near the boundary. This is shown by comparing the volume of a sphere with radius r to that of an interior sphere of radius $r - \varepsilon$, and noting that for arbitrarily small $\varepsilon > 0$ the appropriate ratio of volumes goes to 0 as dimensionality goes to infinity, indicating that essentially none of the volume resides in the interior sphere. That is, "high-dimensional space is mostly empty", which in turn suggests that required sample size for fixed performance grows (rapidly) with dimension. (See also Silverman (1986), Table 4.2.)

Jain et al. (2000) discusses another aspect of the curse, first described by Trunk (1979). It is shown that in the simple case of two d-dimensional multivariate normals with equal (known) identity covariances, known priors $\pi_j = 1/2$, and means

$$\mu_j = (-1)^j \left[1, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{3}}, \ldots, \frac{1}{\sqrt{d}}\right]'$$

for classes $j = 1, 2$, the probability of error for the linear classifier — the classifier which labels an observation as belonging to the class associated with the nearest of the two class-conditional sample means — goes to 0 as $d \to \infty$ if the means are known, but this probability of error converges to $\frac{1}{2}$ if the means must be estimated from any training sample of (arbitrarily large but) fixed size. In other words, adding variates that each decrease the Bayes error can actually increase the classification error when estimates must be used rather than the (unknown) truth.

2.3. Classifiers. Assume for simplicity that the class-conditional probability density functions f_j exist. Then any *density estimator* $\widehat{f}_j$ yields a plug-in classification rule:

$$\widehat{g}(x) = \arg\max_j \widehat{\pi}_j(\mathcal{D}_n)\widehat{f}_j(x; \mathcal{D}_n).$$

For iid training data the class conditional sample sizes, $\widehat{\pi}_j$, are consistent estimators for the priors; if in addition a density estimator is employed for which $\widehat{f}_j \to f_j$ in L_1 or L_2 a.s., for instance, then $L(\widehat{g}|\mathcal{D}_n) \to L^\star$ a.s.

Density estimation comes in two basic flavors, parametric and nonparametric. (We categorize "semiparametric" with nonparametric for the purposes of this discussion.) Parametric density estimation assumes that a parameterized functional form for the class-conditional densities f_j is known and focuses on estimating the (few) unknown parameters. Nonparametric methods, on the other hand, make no such parametric assumption. Parametric density estimation is an easier problem — rates of convergence are faster, for example — due to the fact that the target is finite dimensional. Of course, if the assumed parametric form is not correct, a parametric approach will not in general yield consistent classification. Nonparametric methods provide a more general guarantee of consistency, at a price of reduced efficiency if indeed a simple parametric form is appropriate. Classical examples of these two categories, which allow for a fruitful "compare and contrast" exercise, are given by finite mixture models (McLachlan and Krishnan (1997)) versus kernel estimators (Silverman (1986)).

Density estimation is, however, quite expensive in high dimensions (curse of dimensionality). Thus, for multivariate feature vectors in particular, there is much interest in developing applicable classification methodologies which somehow reduce this cost. One approach involves preprocessing to yield reduced dimensionality without seriously degrading classification performance. Thus, one might choose a projection $\mathbb{P} : \Xi \to \mathbb{R}^{d'}$, where $d' = 1$ or 2, say, and consider classification, as above, using $[(\mathbb{P}(X_1), Y_1), \ldots, (\mathbb{P}(X_n), Y_n)]'$ as the transformed training data. See, for instance, principal component analysis, independent component analysis, linear discriminant analysis, and projection pursuit. These techniques can be found in standard multivariate statistics texts such as Seber (1984), Mardia et al. (1995), Johnson and Wichern (1998), and in pattern recognition texts such as Fukunaga (1990), Duda et al. (2000), and Hastie et al. (2001).

Consideration of the maxim “classification is easier than density estimation” suggests that instead of trying to estimate the probability densities, one might choose to estimate the decision region directly. This, too, can be done parametrically or nonparametrically.

The simplest decision region is a linear one, and several methods involve either estimating the best linear separator of the data or extending to piecewise linear discriminators. See for example Sklansky and Wassel (1979).

A popular nonparametric method is the nearest neighbor classifier (and its extension, the k-nearest neighbor classifier). The idea is simple, yet powerful: choose the category associated with the nearest element of the training set. Given a training set $\mathcal{D}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}'$, the nearest neighbor classifier g_{nn} is defined to be

$$g_{nn}(x; \mathcal{D}_n) = Y_{\arg\min\limits_i \{\rho(x, X_i)\}},$$

where $\rho : \Xi \times \Xi \to [0, \infty)$ is a distance function. This classifier has been studied widely — “simple rules survive!” and is a standard against which new classifiers are often tested.

It is well known that the nearest neighbor rule has asymptotic error bounded above by $2L^\star$. This means that if the classes are strictly separable, so that $L^\star = 0$, then the nearest neighbor classifier is consistent.

The k-nearest neighbor classifier is an obvious extension. Rather than considering only the nearest observation, consider the k nearest elements of the training set. A simple vote is taken amongst the classes. (More complicated voting schemes have been investigated.)

Denoting the k-nearest neighbor classifier by g_k, the following theorem of Stone (1977) establishes the universal consistency of this classifier.

THEOREM. *Given iid training data $\mathcal{D}_n$, if $k \to \infty$ and $k/n \to 0$ then*

$$EL(g_k; \mathcal{D}_n) \to L^\star$$

for all distributions.

Many other classifiers have been, and continue to be, developed. We argue, however, that for high-dimensional problems the choice of classifiers is not the most pressing problem. Rather, dimensionality reduction is the fundamental determining aspect of classification performance in high dimensions.

2.4. Misclassification rate estimation. In order to assess how good a classifier is, or to compare classifiers, we would like to know the misclassification rate (probability of misclassification) L. Unfortunately, knowing the exact value of L requires knowledge of the (unknown) class-conditional distributions. Therefore, an important issue in pattern recognition is the estimation of the misclassification rate.

One method for misclassification rate estimation is called the training/test set method: one selects a training set from which to build the classifier, and holds out an independent test set (for which the class labels are also known) upon which to evaluate the classifier. This unbiased holdout estimate of classification performance is denoted $\widehat{L}_n^m$ where n observations are used in training and m observations are used in testing. Analysis is easy: $m\widehat{L}_n^m$ is the sum of independent Bernoulli random variables, and hence follows a Binomial$(m, L(g|\mathcal{D}_n))$ distribution. A problem with this approach is that it requires the collection of additional labelled data beyond that which is used to build the classifier. Labelled data can be expensive, and one might want to use all the available labelled data for training, under the assumption that this will yield a better classifier.

The method in which one uses all the labelled data to build the classifier and then uses the same data to test the classifier is called *resubstitution*, denoted $\widehat{L}^{(R)}$. The resubstitution error rate can sometimes be useful in the analysis of classifiers, but obviously yields a biased (optimistic) estimate of the error.

An improvement on the resubstitution method, with some of the flavor of the training/test method, is *leave m-out cross-validation*, denoted $\widehat{L}_n^{(m)}$. In this, m observations are withheld from a training set of size n and are subsequently used to test the resultant classifier. This is repeated with the next m observations, until all observations have been in a test set (each observation is used in only one test set). If $m = 1$, this is simply referred to as cross-validation. For a discussion of the relative merits of various methods for estimating misclassification rate, see Devroye et al. (1996) or Ripley (1996).

2.5. Clustering. In the unsupervised case, we have available to us feature vectors $\mathcal{X}_n := \{X_1, \ldots, X_n\}'$, with no class labels available. The goal is to *cluster* these data in such a way as to provide clusters $C_k \subset \mathcal{X}_n$, $k = 1, \ldots, K$ which correspond to some (interesting? useful?) unobserved class labels. Clustering is obviously a more difficult problem than classification. However, clustering is a likely candidate for the exploitation subsystem in some ISP applications.

Clustering can be viewed as the discovery of latent classes within the data. The clusters correspond to classes that were not identified by the collector of the data. These can represent, for example, different variants of a disease in a medical application, previously unidentified subspecies in a biological application, or different types of vehicle in an image processing application.

Unlike classification, clustering *per se* is not well posed. Before proceeding, one must define (implicitly or explicitly) a definition of cluster. Different definitions lead to different clusterings, and without *a priori* information, there is little reason to select one clustering over another. Thus, clustering depends fundamentally on the underlying cluster model.

A further distinction is that clustering requires a determination of the number of clusters. This can be done *a priori*, but usually it is done interactively, either through presentation of potential classes to the user, or through some testing procedure on the model. Thus, clustering combines all of the hard questions in statistics: model selection, model building and model assessment.

3. Integrated Sensing and Processing

The smooth functioning of industry, the government, and even our individual day-to-day activities increasingly relies on a broad spectrum of sensing systems keeping a vigilant eye (ears, nose, etc.) on myriad complex environments and tasks. We are becoming accustomed to the benefits of sophisticated sensing/exploitation systems, ranging from the CT scanners and magnetic resonance imagers that our doctors may inflict upon us, all the way to the suite of radars, thermal imagers, accelerometers, gps, and chemical sensors which some modern cars carry. (Progress.) Moreover, vast quantities of sophisticated sensor data is readily obtained for perusal in the comfort of one's home: large quantities of imagery from webcams, surveillance cameras, hyperspectral sensors, synthetic aperture radars (SAR), and X-ray astronomical data, to name only a few types, can all be quickly accessed on the internet.

The growing complexity and volume of digitized sensor measurements, the requirements for their sophisticated real time exploitation, the limitations of human attention, and increasing reliance on automated adaptive systems all drive a trend towards heavily automated computational processing of the flood of raw sensor data in order to refine out essential information and permit effective exploitation. Complex computational tasks like image formation and enhancement, feature extraction, target detection, classification, intelligent compression, indexing, and operator cueing contribute substantially to the successful operation of the ubiquitous sensing systems essential for our modern technological society.

A generic sensor system may be viewed as a machine for converting information about an object or situation through various representations. The information is initially carried in physical fields (for example, light waves entering a camera lens), transduced into a digital representation (such as the pixels of a grayscale image), which may be computationally manipulated (contrast enhanced for example), and, in many cases, converted to concentrated symbolic information (such as the identification of a particular person standing before the camera). A cartoon model of the generic sensor system is depicted in Figure 1 with the feedforward flow of information from stage to stage indicated by the horizontal arrows. Each subsystem in the figure performs its specific transformation of information in its turn, from physical fields to digital representation in the physical layer, with digital manipulations and enhancements in pre-processing, and finally exploitation to extract high level content. Digital processing generally begins on a pixel array "thrown over the fence" from the physical layer. There is generally little direct feedback from the processing layers to the physical layer that would enable a rapid adaptation of that subsystem's behavior on the basis of discoveries or requirements of processing layers. In consequence, the physical layer typically measures a rather fixed representation of the physical fields, and the digital processor endeavors to extract useful information out of this by computational processing.

Over the last 40 years the need for for effective computational processing and exploitation of digitized sensor data has been met by advances in algorithms from Digital Signal Processing (DSP) and statistical pattern recognition. These advances have combined the power of applied mathematics with the growing precision, stability, throughput, and easy availability of digital processors in an attempt to meet the growing challenges posed by modern applications. One big impact of these advances on sensor systems is the decoupling into the subsystems described previously: physical sensor layer, digital processor layer, digital/symbolic exploitation layer. This represents a significant transformation of sensor/exploitation systems from those of previous times, when exploitation tasks were not automated, and only rudimentary signal processing was performed directly on sensor measurements in the analog domain. Within the current division of labor, analog manipulation is limited to the first stages of the physical sensing, whereas recent computational mathematical developments in DSP and pattern recognition naturally concern the digital processing and exploitation layers almost exclusively.

Recent DARPA sponsored reviews of trends in sensor systems have suggested that the growth of computational complexity in sensor systems networks is quickly becoming a hard limit to scale-up through the concomitant growth of costs of hardware and software, power consumption, and specialization. As sensor data volume and dimensionality grows, computational loads appear to be outstripping the steady Moore's law growth of processor power and the sporadic algorithmic breakthroughs in throughput. One response to this is DARPA's Integrated Sensing and Processing (ISP) program, which attempts to meet this challenge by leveraging mathematical advances across *all* components of a sensing system. ISP seeks examples of sensing systems for which it is possible and advantageous to jointly optimize traditionally the decoupled subsystems of a sensor system. This contrasts sharply with standard approaches which independently optimize subsystems such as the physical layer (sensor head), and the various computational processing layers.

ISP begins with the observation that the main impact of mathematical developments for sensor systems in recent times has been in the processing and exploitation layers, where the ability to computationally adapt mathematical representations and transformations of digital data in real time enable the discovery and exploitation of structure hidden in raw sensor output. Similar but largely untapped opportunities now exist in a current generation of digitally controllable sensor heads for a broad spectrum of phenomena, suggesting new capability to adaptively sense features more informative than pixels.

To realize this capability will require effective mathematical optimizations and control strategies which intelligently integrate currently disjoint tasks of sensing and computation. This promises immediate benefit of "load balancing" between sensor head and processing, with lower signal processing burden while greatly improving the quality and information concentration of the measurements. Carrying on with this idea, ISP contemplates "back end" functions such as classifier algorithms playing an active role in dynamic control of their sensor inputs; in effect playing a mathematically optimal game of "20 questions" through tailored sensor queries suited to the task at hand and what is known or suspected up to the present time. In the new picture of a sensor system, the components have overlapping functionality and communicate data and control in an all-to-all load balanced network.

In this paper, we demonstrate several simple "proof-of-concept" examples of ISP, in which the exploitation subsystem feeds back to the sensor information on what next to sense, based on the determination of the exploitation (classifier) on the current data. Thus, based on preliminary classification of what has been observed, the sensor changes what it is collecting and how it is processing the observations. Again we refer to the cartoon presented in Figure 1. Traditionally, a sensor collects measurements which are processed in some manner and fed to a classifier. The classifier renders its decision and some action is taken based on this decision. This traditional flow is indicated by the horizontal arrows. In adaptive sensors a sensor-preprocessor feedback loop may be present. In the full ISP scenario, the classifier also modifies the set of measurements to be sensed based on exploitation-level feedback. Thus, based on analysis done in the different subsystems, sensor adjustments are fed back to the sensor to improve the overall performance of the system without adversely impacting the overall throughput.

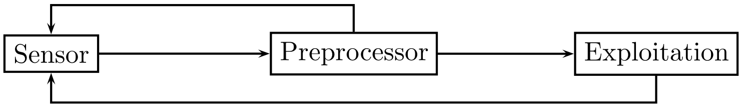

Figure 1. Integrated Sensing and Processing (ISP). The initial sensor measurements are processed in the preprocessor. This may indicate adjustments to the sensor (top arrow) — for example, to improve signal to noise ratio. Preliminary classification results at the exploitation stage suggest changes to the sensing, which information is also fed back to the sensor (bottom arrow).

One analogy for the ISP is a human doctor, viewed as an adaptive sensor/exploitation system. The doctor collects preliminary information, temperature, blood pressure, etc. Then, based on these measurements and external information (for example, information about the outbreak of a plague), the doctor selects new measurements to collect in order to improve or confirm the preliminary diagnosis. This can be viewed as adjusting the sensor to collect different or more precise information, based on a preliminary classification from the exploitation subsystem. Similarly, a hyperspectral sensor might adjust the spectral range of the sensor based on preliminary indications from the classifier of the potential class of the observed object.

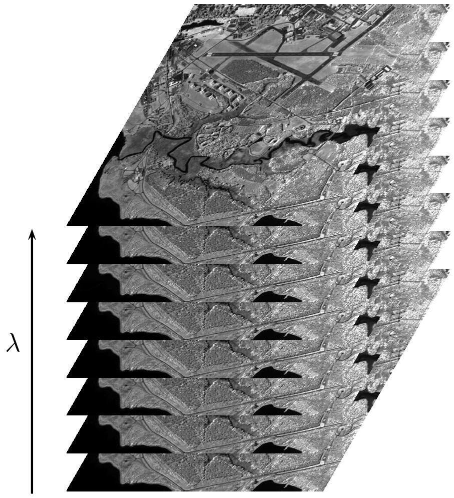

Figure 2. Illustration of a hyperspectral data cube. The cube consists of spatial images (bands) taken at different wavelengths λ.

The ISP approach will be illustrated in the following sections with a pedagogical example and two experimental applications. These illustrations will demonstrate that for some simple but perhaps realistic situations the ISP idea of utilizing information obtained in the classification subsystem to drive sensor parameters can improve the overall performance.

4. Experiment: Hyperspectral Data Cube

For this experiment we have obtained from Naval Space Command a HyMap hyperspectral data set — imagery of the airport at Dahlgren, Virginia (Figure 2). The data consist of 126 images, each one representing the appearance of the scene in light which lies in a narrow spectral band. These bands are obtained throughout the visible, near infrared, and short wave infrared range. Equivalently, we can think of the data as a collection of spectra indexed by the spatial locations in the scene. Spectral imagery data of this sort can provide information about the spatial structure and chemical makeup of the objects within the scene of regard, and is being exploited for problems of detection and identification in a diversity of settings, ranging from biomedicine to defense.

Hyperspectral data gives very fine spectral resolution, but this is not always an advantage. Obviously hyperspectral data is very high-dimensional compared to multispectral imagery, which is similar in concept but comprised fewer, coarser spectral bands. One must be concerned with the curse of dimensionality in the statistical pattern recognition tasks applied to hyperspectral data. Moreover, the large data sets produced by hyperspectral imagers can also lead to significant computational and communication challenges, particularly for time-critical applications. Furthermore, the narrow spectral range of the hyperspectral bands mean that one must collect light for some time before obtaining enough photons in a given band to produce an image with reasonable signal-to-noise ratio. A multispectral sensor with fewer bands would offer coarser spectral resolution but could offer better time resolution, lower dimensional data, and less overall data burden than a hyperspectral sensor. A multispectral sensor with tunable bands could potentially offer some of the benefits of both worlds.

To explore this possibility, we used the more than 100 bands of the HyMap hyperspectral data set as the basis for simulation of a two-band ISP sensor system in which the two are chosen adaptively. For the purposes of this experiment, 6 bands with high noise were removed and 120 bands are used to give an indication of the distribution of photons over wavelength. The coarse bands of the ISP sensor are each the result of a Gaussian filter applied to the 120 band HyMap spectrum. That is, for each spatial location, a weighted sum of the the spectral intensities multiplied by the amplitude of a Gaussian with mean μ_λ and standard deviation σ_λ is returned. Thus the sensor has four adjustable parameters: the spectral means and standard deviations of the Gaussian filters.

Pixels were selected from the image and classed as corresponding to one of 7 classes, using ground truth based on a visit to the site. The 7 classes are: runway, pine, oak, grass, water, brush, swamp. A training set of 700 observations (100 from each class, selected randomly) was chosen, and the remaining (14,048) observations were designated a test set.

The experiment simulates an adaptable sensor which operates as follows. Initially the sensor collects information about the scene in two pre-specified bands (the factory setting), simulated by applying the two Gaussian windows to the HyMap data with fixed initial filter parameter settings. A classifier examines the two band data for each pixel and indicates its coarse classification in the form of the most likely (at most three) classes to which it may belong. Given the classes that this first classifier identifies as contenders, the sensor adjusts its filter parameters to collect new two band data optimized for the task of refining the initial classification by discriminating among the short list of candidates selected in round one. See Figure 3. Thus, the overall sensing and classification takes place in multiple stages with feedback to the sensor to improve the results. The classifiers must be trained and optimized; therefore for all stages, the training data has been split into two equal subsets, with one set used in classifier construction and the other used to estimate the performance of the classifier. More precisely:

Stage 1. We employ a 7-nearest neighbor classifier as the initial coarse-grained classifier. For each observation presented to it, the labels of the top three most likely classes (of the seven defined above) are returned. The filter parameters defining the two bands of the sensor are selected so as to maximize the empirical probability that this classifier places the correct class amongst the top three.

These parameters, along with the 7-nearest neighbor classifier defined by the full training set, constitutes the initial sensor/classification system. This provides the "factory setting" of the system.

Stage 2. For each of the $\binom{7}{3}$ "superclasses" (combinations of 3 candidate classes), filter parameters are selected which optimize the classification of an observation drawn from this superclass, narrowing down its classification to just one of these 3 candidates. That is, we optimize to maximize the probability that an observation is assigned to the correct class given the data available for the 3 class "superclass" identified for that observation in stage 1. The classifier applied to the sensor features tuned to a given superclass is a 1-nearest neighbor classifier based on the training data restricted to the 3 candidate classes of that superclass.

Again, performance is evaluated using the split training set, not the independent test set. The filter parameters selected for each combination of classes will be used to tune the sensor for the best possible discrimination when initial classification of a test observation indicates that particular combination of classes constitutes the candidate set.

Stage 3. The overall classifier is tested as follows. For each observation in the test set, the initial "factory setting" filter parameters are used to obtain the initial two sensor features. The 7-nearest neighbor classifier is evaluated on these initial features. Generally this will return the three leading candidate classes for the observation. In the event that all 7 nearest neighbors are labelled with the same class, unanimity is viewed as decisive and the test observation is classified accordingly without further ado. Otherwise, the filter parameters appropriate to the candidate set of classes are used to adapt the sensor and produce a new feature vector. This new feature vector is passed to the appropriate nearest neighbor classifier, which renders its decision.

The results of this experiment indicate that this optimization which includes feedback from the exploitation subsystem can yield significant performance improvement. The initial classifier places the true class of the test observation into the top three classes 94.15% of the time. This places a lower bound on the possible performance of the overall system at $\widehat{L}_{LB} = 0.0585$. Using a nearest neighbor classifier on these features produces an error of $\widehat{L}_{nn} = 0.1844$. (If instead of optimizing the parameters for the top-3 classifier we optimize for the nearest neighbor classifier we obtain an error of $\widehat{L}_{optnn} = 0.165$.) Our two-stage classifier, which adjusts the sensor based on a preliminary classification as suggested by the "feedback loop" in Figure 1, has an error of $\widehat{L}_{isp} = 0.101$. Thus this experiment demonstrates a significant improvement due to altering sensor parameters based on classification-specific feedback. Notice that we are simulating the effect of the Gaussian filter feature extraction; if implemented in a sensor system, we would expect the classification performance to be even better due to integration gains inherent in observing the spectral features directly.

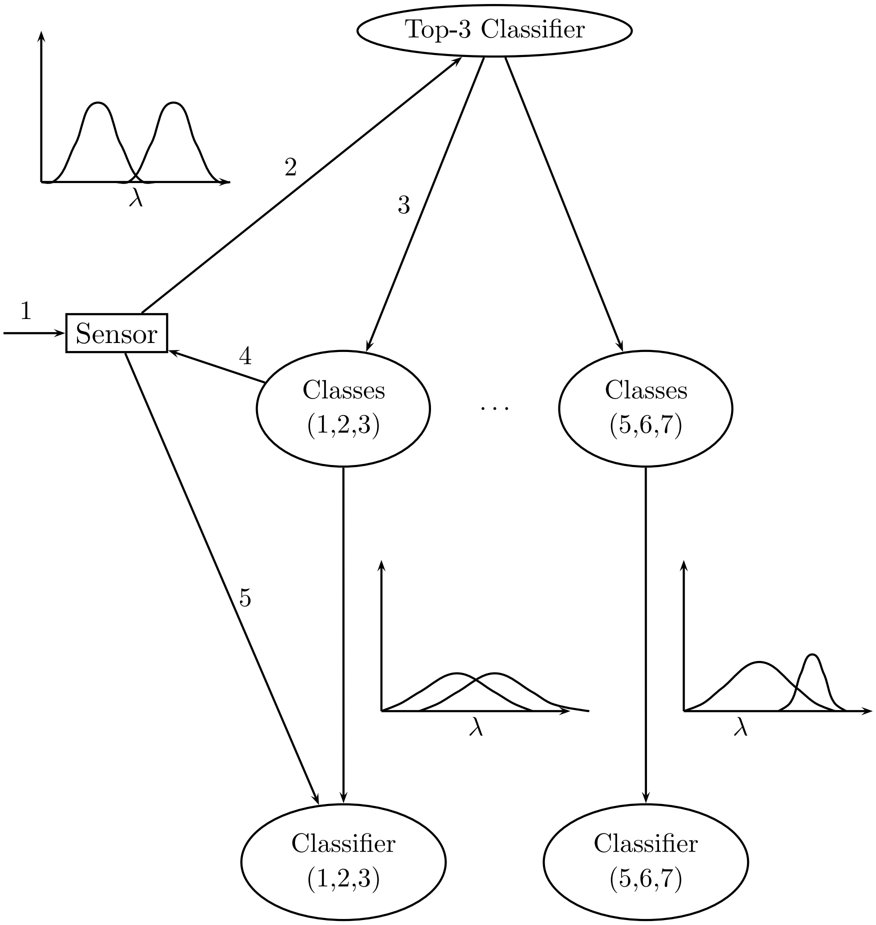

Figure 3. Illustration of the hyperspectral experiment. First, the sensor collects the default bands (1) and a classifier determines the top three classes most likely to contain the true class (2). This determines the new bands to sense (3), which is fed back to the sensor (4). The sensor collects the appropriate bands, which are passed to the ultimate classifier (5).

5. Pedagogical Example: Multivariate Time Series

As a pedagogical example of ISP, consider a case in which each observation consists of a multivariate time series (this sort of data is rather common). For each entity under investigation, the sensor is capable of observing any of $d > 1$ time series ("bands") on a time interval $[0, T]$ at a maximum resolution r_{max} — that is, at equally-spaced times $t_1 = T/r_{max}, t_2 = 2T/r_{max}, \ldots, t_{r_{max}} = r_{max}T/r_{max} = T$. However, sensor and/or channel constraints dictate a maximum throughput for each observation of $\tau < d \cdot r_{max}$. This is a reasonable simplified model of constraints which might imposed on a real systems by limitations of sensor power, available communications bandwidth, computational power, etc.

We want to perform feature selection based on exploitation-level considerations, but the exploitation subsystem cannot have access to all potential features simultaneously. We assume that the sensor/processor subsystem is capable of adapting to subsample each band at a band-specific resolution $r_b < r_{max}$ (with $b \in \{1, \ldots, d\}$) — that is, at equally-spaced times $t_1 = T/r_b, t_2 = 2T/r_b, \ldots, t_{r_b} = T$. (The direct subsampling considered here is done without any filtering of the continuous time input, and may introduce aliasing; we shall see that ISP improvement is nonetheless possible.)

Given a training sample $\mathcal{D}_n$ of entities with known class labels (class-conditional training sample sizes n_j for $j \in \{1, \ldots, J\}$ with $\sum_{j=1}^{J} n_j = n$) the goal is to optimize, based on classification performance, over the collection of band-specific resolutions. That is, we seek

$$\vec{r}^{*} := \arg \min_{\vec{r} \in \mathcal{R}_\tau} L_{\vec{r}}(g|\mathcal{D}_n)$$

where $L_{\vec{r}}(g|\mathcal{D}_n)$ denotes the probability of misclassification for classifier g trained on training sample $\mathcal{D}_n$ which has been subsampled in accordance with resolutions $\vec{r}$ and, for $c > 0$,

$$\mathcal{R}_c := \left\{ \vec{r} = [r_1, \ldots, r_d]' \in [0, r_{max}]^d : \sum_{b=1}^{d} r_b \leq c \right\}.$$

Thus $\mathcal{R}_\tau$ is the collection of band-specific resolutions satisfying the throughput constraint τ.

However, since the exploitation subsystem never sees all the dimensions simultaneously, this optimization must be performed iteratively. That is, we begin with an initial sensor setting (say uniform allocation of resolution, $\vec{r}^{1} = [\tau/d, \ldots, \tau/d]'$) and obtain some measure of which bands are useful for the classification task at hand. This information is provided to the sensor/processor subsystem, and the resolution is increased for the more useful bands and decreased for the less useful bands. (We operate here under the guiding principle that higher resolution for bands with discriminatory information is likely to yield an improvement in classification performance. For this version of ISP to work — as opposed to yielding random search — some such guiding principle must be present to allow the sensor/processor subsystem to choose which measurements to make based on feedback from the exploitation subsystem.)

Let $L^1 := L_{\vec{r}^{1}}(g|\mathcal{D}_n)$ represent the mis-classification performance using features at the initial choice of resolutions, $\vec{r}^{1}$. The (penalized) feature selection in the first iteration,

$$\vec{r}^{1*} := \arg \min_{\vec{r} \in \mathcal{R}_\tau} L_{\vec{r}}(g|\mathcal{D}_n) + \lambda \sum_{b=1}^{d} r_b$$

yields performance $L^{1*} := L_{\vec{r}^{1*}}(g|\mathcal{D}_n)$. We expect, if d is large and the number of bands with significant discriminatory information is small, that $L^{1*} < L^1$. This expected improvement is due to the fact that this feature selection represents dimensionality reduction and, in high dimensions with finite training data, dimensionality reduction done properly can yield superior performance due to the curse of dimensionality. (Recall the Jain–Trunk example.)

A simpler version of this feature selection is to perform a band-by-band analysis to determine which bands are useful and which bands are to be discarded. This can be accomplished by considering the special unpenalized "all or nothing" choice of bands:

$$\vec{r}^{1*} := \arg\min_{\vec{r} \in \tilde{\mathcal{R}}'_\tau} L_{\vec{r}}(g|\mathcal{D}_n)$$

with

$$\tilde{\mathcal{R}}'_\tau := \{\vec{r} = [r_1, \ldots, r_d]' \in \{0, \tau/d\}^d\}.$$

At this stage, those bands b for which $r_b^{1*} = 0$ are to be discarded, with the newly-available channel capacity to be evenly allocated among those bands which have been deemed useful. Thus $\vec{r}^2 = [r_1^2, \ldots, r_d^2]'$ where

$$r_b^2 = I\{r_b^{1*} > 0\} \cdot \tau / \textstyle\sum_\beta I\{r_\beta^{1*} > 0\}.$$

Finally, we define $L^2 := L_{\vec{r}^2}(g|\mathcal{D}_n)$. If our guiding principle — in this case, that higher resolution will increase the discriminatory information in the useful bands, then we expect that $L^2 < L^{1*}$.

Of course, the probability of misclassification is not generally available for use in our optimization objective. Using the available training data $\mathcal{D}_n$ we can, for any given $\vec{r}$, obtain an estimate $\widehat{L}_{\vec{r}}(g|\mathcal{D}_n)$ of the probability of misclassification. Thus we can, in principle, seek

$$\widehat{\vec{r}^*} := \arg\min_{\vec{r} \in \mathcal{R}_\tau} \widehat{L}_{\vec{r}}(g|\mathcal{D}_n).$$

Alternatively, some appropriate surrogate may be employed. For instance, a simple classifier g — a classifier for which $\widehat{L}_{\vec{r}}(g|\mathcal{D}_n)$ is readily available — can be used in the optimization. Then a more elaborate classifier g' can be used for the ultimate exploitation. This surrogate approach will be considered in the sequel. Note, however, that when exploitation means classification, as it does herein, appropriate surrogates will likely still require class label information and may need to reside at the exploitation subsystem — on the opposite side of the channel throughput constraint from the sensor/processor subsystem.

We consider for illustration the case in which each class j, band b process is autoregressive. That is, the i-th observation $X_{j,b,i}$, $i = 1, \ldots, n_j$, is given by an (independent) autoregressive $AR_{j,b}(p)$ process of order $p \geq 1$;

$$X_{j,b,i}(t_k) = \sum_{l=1}^{p} \alpha_{j,b,l} X_{j,b,i}(t_{k-l}) + \varepsilon(t_{j,b,i,k})$$

for $t_k \in \{\ldots, -2T/r_{max}, -T/r_{max}, 0, T/r_{max}, 2T/r_{max}, \ldots\}$, where the $\varepsilon(t_{j,b,i,k})$ are iid normal$(0, \sigma_\varepsilon^2)$. We write $\vec{\alpha}_{j,b} = [\alpha_{j,b,1}, \ldots, \alpha_{j,b,p}]'$ to denote the class-specific, band-specific time series parameter vector. (Recall that a requirement for stationarity yields a constraint on $\vec{\alpha}_{j,b}$.)

In this case, no purely signal processing considerations will allow for the determination of which bands/resolutions are to be preferred. This determination must be made based on feedback from the exploitation module which is in turn based on an analysis necessarily taking into account the class labels — classification performance analysis or some appropriate surrogate.

Maximum likelihood estimates of the parameters $\vec{\alpha}_{j,b}$ can be obtained based on observations of the training entities. These estimates are consistent and asymptotically normal (Anderson (1971)). Thus the training sample provides for an asymptotically Bayes optimal classifier.

Furthermore, this provides for a reasonable surrogate. For each band b an hypothesis test of $H_0 : \vec{\alpha}_{1,b} = \vec{\alpha}_{2,b}$ against the general alternative can be performed using Hotelling's T^2 test statistic (Muirhead (1982)), for instance. Those bands for which the null hypothesis is rejected at some specified significance level are considered to be "useful" for discrimination. The consistency of the hypothesis test employed implies that, in the limit, good bands will not be discarded while most bands with no discriminatory information will be discarded. For instance, for $d = 25$ with exactly five of the bands useful for discrimination, testing at the 0.05 level of significance will be expected to reject for 19 of the 20 useless bands while rejecting for all five of the useful bands (as the estimates $\widehat{\vec{\alpha}}_{j,b}$ approach their asymptotic distributions). It follows that $L^{1\star} < L^1$ for large T.

More specifically, for the two class, two band AR(1) case ($p = 1$, $J = 2$, and $d = 2$), consider $T = 1$, $r_{max} = 100$, and initial sensor settings of $r_b = 50$ for $b = 1, 2$ ($\vec{r}^1 = [50, 50]'$). Let the class $j = 1$ model be specified by $\alpha_{1,1} = \alpha_{1,2} = 0$; similarly, let the class $j = 2$ model be specified by $\alpha_{2,1} = 0$ and $\alpha_{2,2} = 0.1$. (For $p = 1$ we drop the superfluous lag subscript l from the parameters $\alpha_{j,b,l}$.) Thus there is no discriminatory information in band $b = 1$, while band $b = 2$ at the highest resolution will allow for optimal discrimination. For these AR(1) processes, a t-test of $H_0 : \alpha_{1,b} = \alpha_{2,b}$ is an appropriate surrogate, and is here employed. To obtain $\vec{r}^{1*}$ we optimize over $\tilde{\mathcal{R}}'_{100}$ via these t-tests, meaning that if exactly one band rejects the null hypothesis we completely eliminate the band which fails to reject and up-sample, to full resolution $r_{max} = 100$, the band which does reject the null hypothesis. Using class-conditional training sample sizes $n_j = 10$, classification performance based on these observations, as measure by a Monte Carlo estimate $\widehat{L}$ based on 50 Monte Carlo replicates of 100 test samples per class per replicate, is

$$\widehat{L}^1 = 0.2184, \qquad \widehat{L}^{1*} = 0.2156, \qquad \widehat{L}^2 = 0.0426.$$

Thus, as designed, the exploitation-based feedback and sensor adaptation yield $\widehat{L}^2 \ll \widehat{L}^1$. As noted above, the consistency of the hypothesis test employed in this example implies that, for large enough class-conditional sample sizes, this empirically observed result can be proved; that is, $L^2 \ll L^1$. (Note that, since $d = 2$ for this case, $\widehat{L}^1 \approx \widehat{L}^{1*}$ is not surprising.)

Regarding the first feature selection, 43 times out of 50 Monte Carlo replicates this selection correctly chose band $b = 2$ ($\vec{r}^{1*} = [0, 50]'$). In five cases both bands yielded rejection in the hypothesis test, in which cases $L^2 = L^{1*} = L^1$. In one case neither band yielded rejection; again $L^2 = L^{1*} = L^1$. In one case band $b = 1$ only — the wrong selection! — yielded rejection; for this one replicate $\widehat{L}^2_{\text{repl}} > \widehat{L}^{1*}_{\text{repl}} > \widehat{L}^1_{\text{repl}}$.

6. Experiment: “Artificial Nose” Chemical Sensor

We consider data taken from a novel chemical sensor/optical read-out system designed and constructed at Tufts University. The fundamental component of this sensor is a solvatochromic dye embedded in a polymer matrix White et al. (1996) which responds to the introduction of a chemical analyte to its environment with a change in its fluorescence intensity. These basic devices can be fabricated in a number of well characterized variants, each responding in some way to particular chemical analytes Dickinson et al. (1996). In general, the devices are cross reactive rather than specific; that is, each will respond significantly to a variety of analytes, although fortunately with differences in the details of the response signature from one analyte to another. By analyzing the responses of several of these devices one may obtain a specific identification in many cases of interest.

For application of these devices in a sensor system, the fluorescence signature must be stimulated and read-out during the exposure of a device to an analyte. For example, a device can be attached to an optical fiber through which laser illumination is provided in order to stimulate the signature fluorescence of that device. The resulting light signal is conducted back through the same fiber for read-out. Typically, an array of devices with their optical fiber readouts will be bundled together to make a sensor. See Priebe (2001) for a discussion of pattern recognition for this kind of sensor.

The Tufts data we study in this section was obtained from a bundle of 19 varying sensors attached to fibers. An observation is obtained by passing an airborne analyte (a single chemical compound or a mixture) over the fiber bundle in a four second pulse, or “sniff.” The information of interest is the change over time in emission fluorescence intensity of the dye molecules for each of the 19 fiber-optic sensors (see Figure 4).

Data collection consists of recording sensor responses to various analytes at various concentrations. Each observation is a measurement of the time varying fluorescence intensity at each of two wavelengths (620 nm and 680 nm), within each sensor of the 19-fiber bundle. The sensor produces observations $X_{j,i,b}(t_k)$ where $b = 1, \ldots, d = 38$ represents the fiber-bandwidth pair $\phi \cdot \lambda$ for fibers

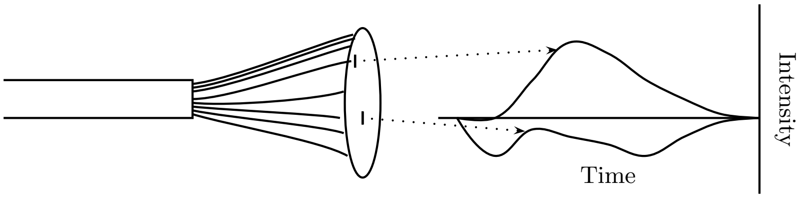

Figure 4. The Tufts artificial nose consists of optical fibers doped with a solvatochromic dye. Reaction of the polymer matrix with an analyte produces photons which are sampled at two wavelengths to produce a response for each fiber. These photons are captured by a CCD device, resulting in a time series of light intensity above (or below) the background intensity. The figure illustrates the response of two fibers sampled at a single wavelength.

$\phi \in \{1, \ldots, 19\}$ and wavelengths $\lambda \in \{1, 2\}$. The index $i = 1, \ldots, n$ represents the observation number. The class label j flags the presence or absence of a chemical of interest, described in more detail below. While the process is naturally described as functional with t ranging over a 20 second interval $[0, T = 20]$, the data as collected are discrete with the 20 seconds recorded at $r_{max} = 60$ equally spaced time steps $t_k = \frac{20}{60}, \frac{40}{60}, \ldots, \frac{1200}{60}$, for each response. Construction of the database involves taking replicate observations for the various mixtures of chemical analytes.

The sensor responses are inherently aligned due to the "sniff" signifying the beginning of each observation. The response for each sensor for each observation is normalized by manipulating the individual sensor baselines. This preprocessing consists of subtracting the background sensor fluorescence (the intensity prior to exposure to the analyte) from each response to obtain the desired observation: the change in fluorescence intensity for each fiber at each wavelength. Functional data analysis smoothing techniques are utilized to smooth each sensor response Ramsay and Silverman (1997).

The task at hand is the identification of an unlabelled odorant observation X. Specifically, we consider the detection of trichloroethylene (TCE) in complex backgrounds. (TCE, a carcinogenic industrial solvent, is of interest as the target due to its environmental importance as a groundwater contaminant.)

In addition to TCE in air, eight diluting odorants are considered: BTEX (a mixture of benzene, toluene, ethylbenzene, and xylene), benzene, carbon tetrachloride, chlorobenzene, chloroform, kerosene, octane, and Coleman fuel. Dilution concentrations of 1:10, 1:7, 1:2, 1:1, and saturated vapor are considered.

We consider the training database $\mathcal{D}_n = [(X_1, Y_1), \ldots, (X_n, Y_n)]'$ to consist of 38-dimensional time series (representing odorant observations) and their associated class labels $Y_i \in \{1, 2\}$ (TCE absent and present, respectively). The database $\mathcal{D}_n$ consists of n_1 observations from class 1 and n_2 observations from class 2. Class 1, the TCE-absent class, consists of $n_1 = 352$ observations; the database $\mathcal{D}_n$ contains 32 observations of pure air and 40 observations of each of the eight diluting odorants at various concentrations in air. There are likewise $n_2 = 760$ class 2 (TCE-present) observations; 40 observations of pure TCE, 80 observations of TCE diluted to various concentrations in air, and 80 observations of TCE diluted to various concentrations in each of the eight diluting odorants in air are available. Thus there are $n = n_1 + n_2 = 1112$ observations in the training database $\mathcal{D}_n$. This database is well designed to allow for investigation of the ability of the sensor array to identify the presence of one target analyte (TCE) when its presence is obscured by a complex background; this is referred to as the "needle in the haystack" problem. This is the database considered in Priebe (2001).

As in our pedagogical autoregressive process example, we consider a throughput constraint. In this case, with $d = 38$ and $r_{max} = 60$, consider a throughput constraint of $\tau = 1140 < d \cdot r_{max} = 2280$. Then $\tau/d = 30$. Let $\vec{r}^1 = [\tau/d, \ldots, \tau/d]' = [r_{max}/2, \ldots, r_{max}/2]'$. With this initial set up we obtain $\widehat{L}^1 = 0.237$. (Probability of misclassification error rates here are obtained via 10-fold cross-validation using the one-nearest neighbor classifier.)

We obtain $\vec{r}^{1*}$ by optimizing over $\mathcal{R}'_\tau$. Actually, this still leaves 2^{38} candidate dimensionality reductions to consider, and so we "sub-optimize"; we calculate $\widehat{L}_b(g|\mathcal{D}_n)$ for each individual band $b = 1, \ldots, d$ and select the "best few". A subset of 12 of the 38 bands are selected based on this criterion, and after this optimization we obtain $\widehat{L}^{1*} = 0.121$.

The best 12 individual bands selected for $\vec{r}^{1*}$ are then upsampled, while the remaining 38 are downsampled. The components of $\vec{r}^2$ are given by

$$r_b^2 = I\{r_b^{1*} > 0\} \cdot r_{max} + I\{r_b^{1*} = 0\} \cdot r_{max}/4.$$

After optimization and feedback adjustment we obtain $\widehat{L}^2 = 0.102$.

We have, as desired, $\widehat{L}^2 < \widehat{L}^{1*} < \widehat{L}^1$. The improvement from $\vec{r}^1$ to $\vec{r}^{1*}$ is dramatic, indicating that the dimensionality reduction employed — although simplistic — was successful. Using $\vec{r}^2$ as opposed to $\vec{r}^{1*}$ yields an improvement of 1.9%. The reduction in misclassification rate is from 134 misclassified to 113 misclassified — 21 observations, or 15.7% of the previously misclassified observations. This improvement obtained by using $\vec{r}^2$ as opposed to $\vec{r}^{1*}$ is statistically significant (McNemar's test).

7. Discussion

We have presented examples illustrating "Integrated Sensing and Processing" (ISP) as a path towards end-to-end optimization of a sensor/processor/exploitation system with respect to its performance in supervised statistical pattern recognition (classification) tasks. The approach we have studied in this paper takes the form of dimensionality reduction in sensor feature space coupled with adaptation of sensor features. These techniques are aimed explicitly at improving an exploitation objective — probability of misclassification — and are necessarily implemented iteratively due to throughput constraints.

We note that the results presented are quite preliminary and only begin exploration of the ISP concept. For instance, classifier adaptation and optimization is certainly an aim in ISP, although we have not pursued this direction in the present paper. Ultimately, ISP seeks to jointly optimize sensor function, digital preprocessing, and exploitation systems, including classifier design; however, it is our belief that this issue is secondary to that of dimensionality reduction for many high-dimensional classification applications.

Dimensionality reduction is fundamentally important for many disparate applications in pattern recognition as well as in other fields including control, modeling and simulation, operations research, and visualization. The topic is the subject of intense research in these various communities, and now becomes a fundamental enabling technology for the new discipline of ISP. In this paper we have considered only very simple dimensionality reduction methodologies, which just begin to indicate the possibilities and implications for integrating sensing and processing. Nevertheless, we feel that the results of these first experiments indicate significant promise for this line of inquiry.

A critically important aspect of the dimensionality reduction strategies considered in this paper is the identification of some guiding principle or heuristic for guiding the sensor/processor subsystem in its choices of which measurements to make based on dimensionality-reduction feedback from the exploitation subsystem. The choice of such a principle is a sensor- and application-specific task. For many multivariate time series scenarios "higher resolution in useful bands" approach taken in this paper seems to be a reasonable principle. This might be extended to include variable resolution in quantization, or in spatial sampling in other sensors. Finding appropriate guiding principle(s)for various important cases of practical interest may perhaps represent the single most important aspect of developing a workable ISP methodology.

References

T. W. Anderson. *The Statistical Analysis of Time Series.* Wiley, New York, 1971.

R. E. Bellman. *Adaptive Control Processes.* Princeton University Press, Princeton, New Jersey, 1961.

L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition.* Springer, New York, 1996.

T. Dickinson, J. White, J. Kauer, and D. Walt. A chemical-detecting system based on a cross-reactive optical sensor array. *Nature*, 382:697–700, 1996.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* Wiley, New York, 2000.

K. Fukunaga. *Statistical Pattern Recognition.* Academic Press, San Diego, 1990.

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

A. K. Jain, R. P. W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

R. A. Johnson and D. W. Wichern. *Applied Multivariate Statistical Analysis.* Prentice Hall, New Jersey, 1998.

K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis.* Academic Press, New York, 1995.

G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions.* Wiley, New York, 1997.

R. J. Muirhead. *Aspects of Multivariate Statistical Theory.* Wiley, New York, 1982.

C. E. Priebe. Olfactory classification via interpoint distance analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:(4):404–413, 2001.

J. Ramsay and B. Silverman. *Functional Data Analysis.* Springer, New York, 1997.

B. D. Ripley. *Pattern Recognition and Neural Networks.* Cambridge University Press, Cambridge, 1996.

D. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization.* Wiley, New York, 1992.

G. A. F. Seber. *Multivariate Observations.* Wiley, New York, 1984.

B. W. Silverman. *Density Estimation for Statistics and Data Analysis.* Chapman and Hall, New York, 1986.

J. Sklansky and G. Wassel. *Pattern Classifiers and Trainable Machines.* Springer, New York, 1979.

G. V. Trunk. A problem of dimensionality: A simple example. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(3):306–307, 1979.

J. White, J. Kauer, T. Dickinson, and D. Walt. Rapid analyte recognition in a device based on optical sensors and the olfactory system. *Anal. Chem.*, 68: 2191–2202, 1996.

Carey E. Priebe
Department of Mathematical Sciences
Johns Hopkins University
Baltimore, MD 21218-2682
United States
cep@jhu.edu

David J. Marchette
Naval Surface Warfare Center, B10
Dahlgren, VA 22448-5100
United States
marchettedj@nswc.navy.mil

Dennis M. Healy, Jr.
Department of Mathematics
University of Maryland
College Park, MD 20742-4015
United States
dhealy@math.umd.edu