Issues in the Identification of Smoke in Hyperspectral Satellite Imagery — A Machine Learning Approach

Mark A. Wolters and C.B. Dean

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/60214

1. Introduction

Observations from earth-orbiting satellites play an important role in the study of various large-scale surface and atmospheric phenomena. In many cases the data collected by such satellites are used and communicated in the form of raster images—three-dimensional data arrays where the first two dimensions define pixels corresponding to spatial coordinates. The third dimension contains one or more image *planes*. A greyscale image, for example, has one image plane, while a color (RGB) image has three planes, one each for the brightness in the red, green, and blue parts of the visible spectrum.

The present work is related to *hyperspectral* images, where the number of image planes is much greater than three. In a hyperspectral image with r planes there is associated with each pixel a set of r data values, each measuring a different part of the electromagnetic spectrum.

The general task of analyzing geographic remote sensing imagery is aptly described by Richards [1] (p. 79):

> *With few exceptions the reason we record images of the earth in various wavebands is so that we can build up a picture of features on the surface. Sometimes we are interested in particular scientific goals but, even then, our objectives are largely satisfied if we can create a map of what is seen on the surface from the remotely sensed data available...*
>
> *There are two broad approaches to image interpretation. One depends entirely on the skills of a human analyst—a so-called photointerpreter. The other involves computer assisted methods for analysis, in which various machine algorithms are used to automate what would otherwise be an impossibly tedious task.*

Here, we will consider methods that are useful for the second approach: computer-assisted photointerpretation. Computer-aided analysis is particularly helpful for hyperspectral images, which contain too many planes to be visualized in a simple human-readable form.

The present work can be viewed as a case study in the application of machine learning approaches to a difficult task in remote sensing image segmentation. The remainder of this section introduces the problem we are addressing, the data we are using, and the modelling approach we will follow. In Section 2, important ideas from the field of classification are introduced in a tutorial format for researchers who might not be familiar with the topic. Those with prior experience in the area may wish to skip the section. Sections 3 and 4 describe the methods used and the results obtained. Sections 5 and 6 provide discussion and conclusions.

1.1. The problem

The application of interest is the automated identification of smoke from forest fires using hyperspectral satellite images. Smoke released from forest fires can be transported large distances and affect air quality over large areas, making it a matter of population health concern. Despite the importance of smoke events, their spatial scale makes them difficult to quantify through direct measurement. Satellite imagery is an alternative information source that could potentially fill a data gap, providing information about smoke over large areas at times of interest.

The work reported here is the first step in a research stream with the ultimate goal of developing a system that can quantify smoke using moderate- to high-resolution remote sensing images covering large geographic areas, and do so with minimal human intervention. If smoke can be quantified through remote sensing image analysis, the resulting data could be used as input to deterministic predictive models of forest fire smoke dispersal, as a validation check for such models, or as an input to retrospective studies of the health impacts of smoke.

Our present objective is twofold: first, to report our current results in developing a classifier for smoke detection, and second, to stimulate other researchers to consider applying similar methods for their own problems in remote sensing image analysis.

1.2. The data

The region of interest in this study covers parts of western Canada and the northwestern United States, and is centered close to the city of Kelowna, British Columbia. It extends from 46.5° to 53.5° latitude, and from -126.5° to -112.5° longitude. Data come from the moderate resolution imaging spectroradiometer (MODIS) aboard the Terra satellite, which provides images with 36 planes covering different spectral bands ranging from the blue end of the visible spectrum (400 nm) to well into the infrared (14 μ m). More information about MODIS can be found in [3, 4].

The Terra satellite follows a polar orbit that allows MODIS to image most of the globe each day, with images captured at mid-morning local time. All data are freely available from the LAADS web data portal [5]. There are numerous data products available, at different levels of processing for different purposes. We used the Level 1B data at 1km resolution, which provides the hyperspectral data in calibrated form corrected for instrumental effects, but without further manipulation. The data are available in chunks called *granules*. Each granule holds the instrument's observations as it passed over a certain portion of the earth's surface during a particular five-minute time interval. If a study region does not happen to be covered by a single granule, it is possible to stitch the data from adjacent granules to cover the region. If the region is large enough, it may be necessary to stitch granules from different orbital passes. In our case, we only used data from time-sequential granules, and not those from different passes, because we found that the smoke and clouds in the scene could change significantly between orbital passes. Because of this it was not always possible to collect complete data for the entire region of interest on every day.

A total of 143 images were collected, one for each day covering the peak dates of the fire season (July 15 to August 31) for the years 2009, 2010, and 2012. Each image is approximately 1.2 megapixels in size, and has spatial resolution of approximately one kilometer per pixel. Images are in plate carrée projection. Any pixel that had data quality concerns (as indicated by error codes in the downloaded data) was excluded from the analysis. The entirety of band 29 was also discarded because of a known hardware failure, leaving 35 spectral bands to be used for classification purposes.

To aid in visualization of the data, an RGB version of each image was produced. Following [6], the RGB images were created by letting bands 1, 4, and 3 fill the red, green, and blue image planes, respectively. First, each of these three bands was run through a saturating linear brightness re-mapping, letting 1 percent of the pixels be saturated at each end of the brightness range. Then, a piecewise linear brightness transformation was carried out on each band, as in the reference.

The resulting RGB images were used for the important task of manually assigning each pixel to either the smoke or nonsmoke class—that is, for specifying what the "true class" of each pixel was. To make this task easier, fire locations (found by comparing bands 22 and 31, as in [7]) were overlaid on the RGB images. While the smoke was sometimes easy to distinguish from the rest of the image, there were also many cases where the choice of true class was quite ambiguous: regions where smoke and cloud were mixed, or regions where the smoke was not highly concentrated, for example. Nevertheless, each pixel in all 143 images was assigned a true class label on a best-efforts basis. The approach to assigning true labels was to assign the smoke class whenever a pixel appeared to have any level of smoke, even a thin haze. The end result was a set of 143 black and white *mask* images corresponding to the hyperspectral ones, with white pixels indicating smoke and black indicating nonsmoke. The complete set of masks comprised 90% nonsmoke pixels and 10% smoke pixels.

As will be shown at the end of this chapter, the difficulty assigning true classes with high confidence is a potentially critical limitation of the analysis. The manual approach to labelling was used nonetheless, since no alternative method exists for identifying smoke pixels across entire images. We note in passing that we have previously obtained some "gold standard" images by request from NASA, and in this case smoke was also identified as hand-drawn regions.

1.3. Modelling approach

The observed images are the product of natural processes that are very complex. From a statistical standpoint, a sequence of remote sensing images covering a particular region of the earth is a spatiotemporal data set with statistical dependence both within and between images. Physically, the presence of smoke in a particular region at a particular time is surely dependent on the characteristics of a particular fire, as well as on meteorological and topographical variables that vary over the region of interest and over time. There is thus ample scope for mathematical complexity in a model used for classification. Some decisions must be made at the outset about which aspects of the problem to include in our classifiers, and which to ignore. As the research is still in its early stages, three simplifying decisions have been made.

First, classification will be conducted based only on the spectral information in the images themselves; no ancillary information (for example, about wind, fire locations, or topography) will be used to aid prediction. This decision was made partly to limit model complexity, but also to ensure that our methods are wholly independent of any physics-based deterministic models (which they might eventually be used to validate). Using only the hyperspectral data also maximizes the applicability of the methods to other image processing tasks.

Second, the focus is on detecting only the presence or absence of smoke. A successful system will be able to classify images on a pixel-by-pixel basis into one of two categories, "smoke" or "nonsmoke."

Third, all pixels and all images are assumed to be independent of one another. While ignoring temporal dependence from image to image does not throw away much information—with images collected at a frequency of once per day, there is little correlation between smoke locations from one image to the next—ignoring spatial dependence within images is clearly making a compromise. Smoke appears in spatially contiguous regions, so knowledge that a certain pixel contains smoke should influence adjacent pixels' probability of being smoke. Nevertheless, spatial association between the outcomes introduces many technical difficulties, so it was not included at this stage of our study.

With these decisions, the smoke detection task becomes a typical *binary classification* or *binary image segmentation* problem, using the data in the 35 spectral bands as predictors. Simplifying the problem in this way is justified in a preliminary analysis. Our goal is to evaluate whether the spectral data contain enough information to allow the smoke and nonsmoke pixels to be distinguished from one another with reasonably high probability. If they do not, there is little to be gained from the added complexity of more sophisticated models; if they do, the simple independent-pixel smoke/nonsmoke model can be extended in a variety of ways to obtain further improvements. Furthermore, it will be seen that despite retreating to a simple model for classification, the problem is still high dimensional, computationally intensive, and challenging.

With these considerations in mind, we use logistic regression for building our classifiers. Logistic regression has convenient extensions for accommodating spatial associations, for handling multiple levels of smoke abundance, and for including additional predictor variables. We anticipate that a final, useful future system will be based on such an extended model.

All analyses presented here were carried out using the free and open source statistical computing software R [2]. An R script demonstrating much of the analysis is available on the corresponding author's website (www.mwolters.com); readers interested in working with the full data set (which is large) can contact the authors by email.

2. Binary classification concepts

Classification is the process of assigning a category (a class label) to an item, using available information about the item. We are interested in binary classification, where there are only two class labels. In our case, the labels are nonsmoke (class 0) and smoke (class 1), the items to be classified are image pixels, and the available information is the content of the hyperspectral image. We say we have "built a classifier" when we have established a rule that tells us how any given pixel in a new image should be classified.

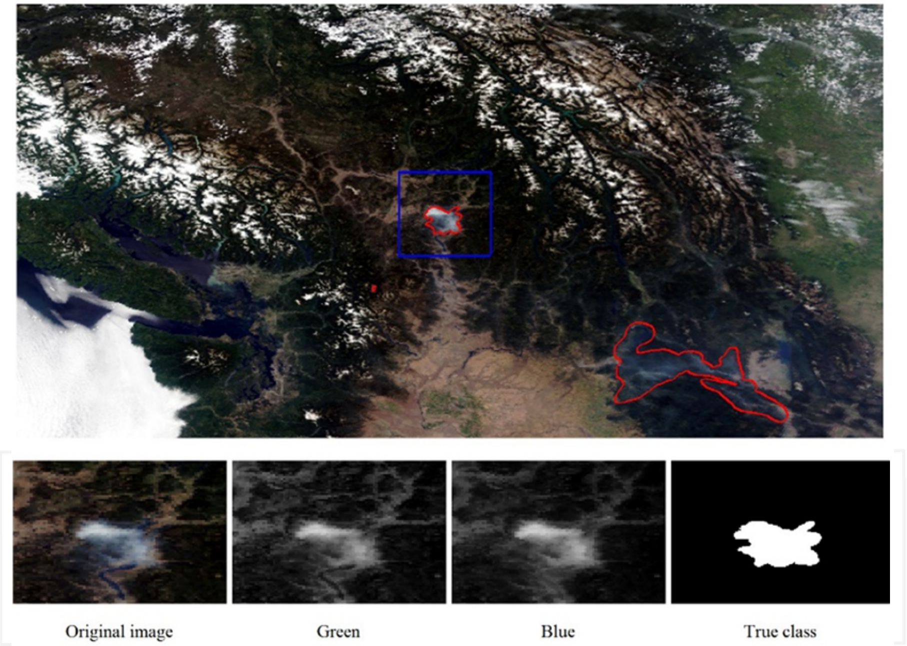

Figure 1. The data used for the example. Top: an RGB image of the study region, with regions of smoke outlined in red. The blue rectangle encloses the pixels that are used for the example. Bottom, from left to right: the RGB sub-image of the region of interest, the green channel, the blue channel, and the mask showing the true smoke (white) and true nonsmoke (black) regions.

Classifier building requires the availability of *training data*—a set of items where the true class labels are known. The reliance on training data is one reason classification is also known as *supervised learning*. One may think of an all-knowing supervisor who tells us the class membership of a subset of our items, but then goes home for the day, leaving us to learn for ourselves how to classify the remaining items. To prevent confusion, note that the alternative problem of *unsupervised learning* (where the wise supervisor never shows up, leaving all class labels unknown) is also known as *clustering*, and—although important in its own right—is not presently relevant.

Classification is a large topic. It is, in fact, the dominant activity in the field of machine learning. Consequently, no attempt is made here to provide a thorough review of the subject. Rather, a single classifier based on logistic regression will be discussed as a means of introducing common themes in classification. The logistic classifier is naturally suited to binary classification problems, and has a relatively simple form with strong connections to linear and nonlinear regression. This classifier will be used throughout the chapter.

Readers interested in further background on classification, and alternative classifiers, have many resources to turn to. The books [1, 8, 9, 10] provide accessible introductions to the topic, and [1] in particular discusses classification and many related topics in the context of remote sensing imagery. Note that while alternative classification methods may have better or worse performance in different situations, most of the important aspects of setting up and solving a classification problem remain the same regardless of the particular method chosen.

2.1. A small example

As an illustrative example, we restrict our attention to a small subset of the study data—a portion of a single image—and work with only the RGB image rather than the full hyperspectral data. The large image in Figure 1 shows the entire study region on the chosen date (and also provides an example of what the color images look like on a clear day). The picture contains two areas outlined in red. These are the areas that were deemed to contain smoke during the masking process. The blue rectangle in the image outlines the set of pixels used for this example. The four smaller images at the bottom of the figure show the example data in more detail: the RGB image, the information in the green channel, the information in the blue channel, and the corresponding mask showing the true classes.

The sub-image used for the example is 150 by 165 pixels (24750 pixels in all) and is centered on a smoke plume. To allow the problem to be visualized in two dimensions, we will consider only the green channel (G) and the blue channel (B) as predictors in our classifier.

2.1.1. Logistic classifier with two predictors

The logistic classifier is based on logistic regression, which is set up as follows. Let the true class (the response variable) of the i th pixel be Y_i, with $Y_i=1$ corresponding to smoke and $Y_i=0$ corresponding to nonsmoke. The true class is modelled as a Bernoulli random variable with $\pi_i=P(Y_i=1)$ being the probability of the smoke outcome. All pixels are assumed to be statistically independent.

Logistic regression models the log-odds of pixel i being smoke (the event $Y_i=1$) as a linear combination of predictor variables (the green and blue brightness values, in this case):

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + \beta_1 G_i + \beta_2 B_i, \tag{1}$$

where G_i and B_i are the green and blue values of the i th pixel, and $\{\beta_0, \quad \beta_1, \quad \beta_2\}$ are the model coefficients. These three coefficients are to be estimated from a set of pixels for which both the responses and the predictors are known. Estimation is done using a weighted least squares or (equivalently) maximum likelihood approach. The process is called *model fitting* or *training*, and software for performing the estimation is readily available.

Once the parameters are estimated, the fitted model can be used to generate predictions for any given pixel, whether or not the response has been observed. Let x_j represent such a pixel, with predictor values G_j and B_j. Plugging G_j, B_j, and the fitted coefficients into the right hand side of (1), the equation can be solved for $\hat{\pi}_j$, the *fitted probability*. This quantity is the estimated probability that pixel j belongs to the smoke class.

The logistic regression model gives us fitted probabilities on a continuous scale from zero to one. To convert the model into a binary classifier, one need only specify a cutoff probability, c. If $\hat{\pi}_j$ is less than c, pixel j will be put into class 0 (nonsmoke), and if $\hat{\pi}_j$ is greater than c, it will be put into class 1 (smoke). We choose $c=0.5$, so that each pixel is put into the class that is more probable under the model.

Returning to the example data, the above procedure was followed using the 24750 chosen pixels and their true class labels as training data to fit model (1). The nature of the resulting fitted model is shown in Figure 2. The figure plots each pixel as a point in the (green, blue) plane. In machine learning, predictor variables are often called *features*, and so this plot considers each pixel in the model's *feature space*. We see that the smoke pixels generally occur at higher values of both blue and green, but that there is overlap between the two classes; the two classes are not completely separable. The fitted logistic regression model allows us to calculate a probability of being smoke for any point in the feature space. The thick line on the plot is the probability 0.5 contour of this probability surface; it is the decision boundary for our classifier with $c=0.5$. The model will classify any pixel above this line as smoke, and any pixel below the line as nonsmoke.

The inset image in the figure shows the classifier's predictions. White pixels in this image indicate pixels estimated to have greater than 50% chance of being smoke. The red outline indicates the boundary of the true smoke region. While most of the pixels are classified correctly, many are not.

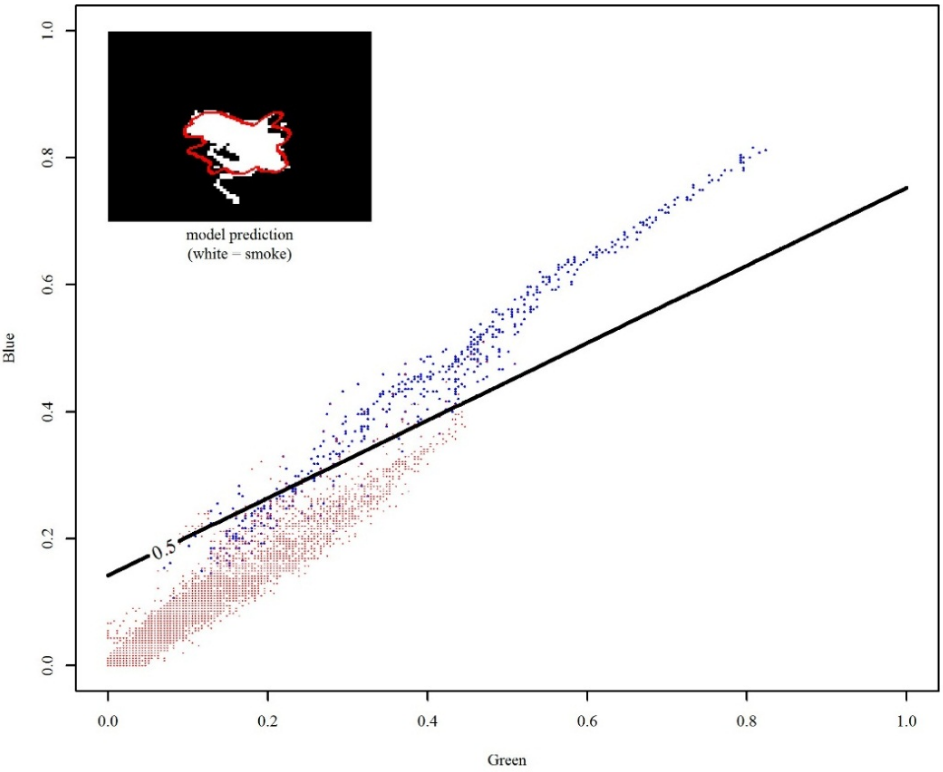

Figure 2. Results of fitting the two-predictor model (G, B) to the example image. Blue points are smoke pixels and red points are nonsmoke. The line on the plot gives the 50% probability line that can be used to discriminate one class from the other. The inset image shows the predicted classes using this model; the red outline in the inset is the boundary of the true smoke region.

2.1.2. Logistic classifier with expanded feature space

The mathematical structure of the previous model ensured that the decision boundary in Figure 1 had to be a straight line. This limited the ability of the classifier to discriminate between the two classes. To make the model more flexible, we can expand the size of the feature space by adding nonlinear functions of the original predictors G and B. For example, we can consider the model

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + \beta_1 G_i + \beta_2 B_i, + \ \beta_3 \mathrm{G}_\mathrm{i}^2 + \beta_4 \mathrm{B}_\mathrm{i}^2 + \\ + \beta_5 \mathrm{G}_\mathrm{i}\mathrm{B}_\mathrm{i} + \beta_6 \mathrm{G}_\mathrm{i}^3 + \beta_7 \mathrm{G}_\mathrm{i}\mathrm{B}_\mathrm{i}^2 + \beta_8 \mathrm{B}_\mathrm{i}\mathrm{G}_\mathrm{i}^2 + \beta_9 \mathrm{B}_\mathrm{i}^3 + \beta_{10} \mathrm{G}_\mathrm{i}^2\mathrm{B}_\mathrm{i}^2, \qquad (2)$$

which includes the original variables G_i and B_i, along with squared and cubed terms (like G_i^2 and G_i^3) as well as products between the original variables taken to various powers (as in G_iB_i and $B_iG_i^2$). Borrowing terminology from industrial experimentation, we call the original variables *main effects* and any terms involving products of variables *interactions*.

The right hand side of model (2) is still a linear combination of various predictor variables, but we have expanded the feature space to ten dimensions. Considered as a function of G and B, the model is able to handle nonlinear relationships between these main effects. In Figure 3 we see the results of fitting this model to the example data. The figure shows the same scatter plot of the data, but now with the 50% contour line for this more flexible model. By adding extra features we can define a decision boundary with more complex shape. The additional shape flexibility of this boundary allows the classifier to correctly assign classes to a greater proportion of the pixels, as seen in the inset prediction image.

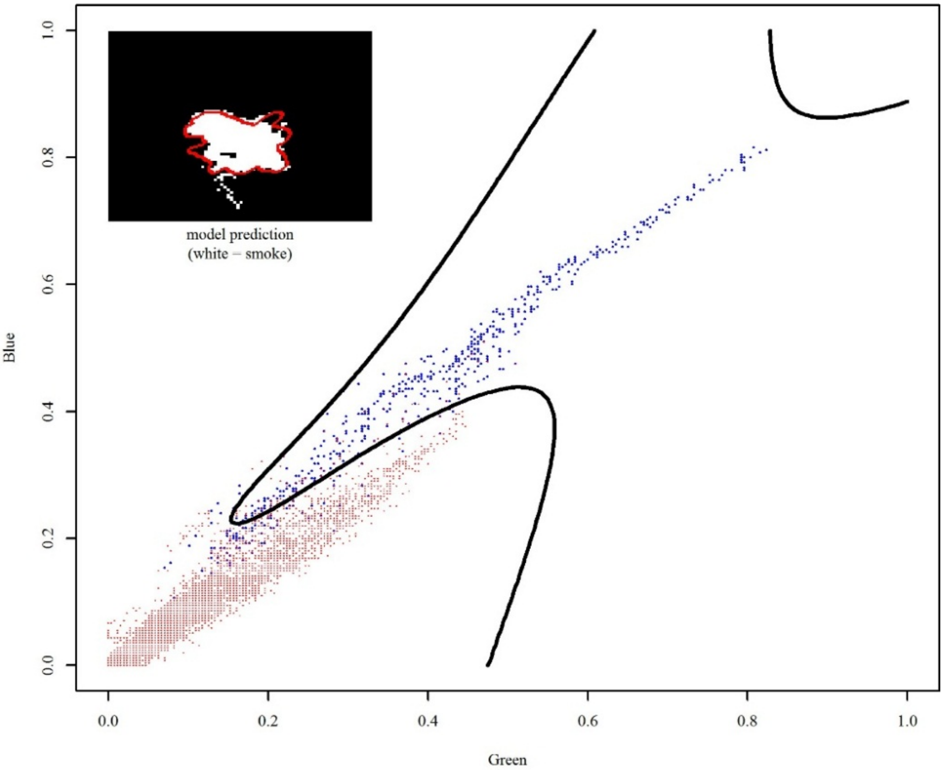

Figure 3. Results of fitting the example data to the 10-predictor model (G, B, G^2, B^2, GB, G^3, GB^2, BG^2, B^3, G^2B^2). The plot is constructed in the same way as the previous figure. In this case the class decision boundary can take a complex non-linear shape.

2.2. Other important concepts

The preceding example might tempt one to believe that simply adding more predictors to the model will always yield a better classifier. This is not true, however, for two reasons.

The first problem with arbitrarily growing the feature space is purely computational. In most problems (and certainly in the present study), the measured main effects are correlated with each other to varying degrees. When expanding the feature space, the variables in the model will increasingly suffer from a form of redundancy known *multicollinearity*: certain predictors can (almost) be written as linear combinations of the other predictors. When the degree of multicollinearity is mild, model fitting will still be possible, but the coefficient estimates can be grossly inaccurate (and can vary greatly from sample to sample). As the problem gets worse, fitting will fail due to the occurrence of numerically singular matrices in the estimation routine.

The multicollinearity problem does not preclude us from considering a large feature space, but it means we cannot include *all* variables from a large feature space in the model. This leads to the problem of *model (feature) selection*: when the number of potential predictors is large, we seek to choose a subset of them that produces a good classifier that is numerically tractable.

When selecting a model from a large collection of correlated predictors, it is important to remember that the coefficient estimate of a particular variable will vary depending on which other variables are included in the model. Further, the best-fitting models of two different sizes need not share their variables in common (the variables selected in the best five-variable model, for example, might not be present in the best ten-variable model). For these reasons it is best to consider the performance of a model as a whole, rather than paying undue attention to coefficient values, statistical significance tests, and the like.

The second problem is more fundamental, and can arise even when multicollinearity is not present. The predictions shown in the previous figures were predictions made on the training data itself; the same data were used both for model fitting and for evaluating performance. This circumstance leads to *overfitting* and poor *generalization* ability: the model fits the training data very well but, because the training data is only a sample from the population, the model's predictive power on new data suffers. When considering increasingly complex models, a point is reached at which additional complexity only detracts from out-of-sample prediction accuracy.

The remedy for overfitting again involves model selection. Because of overfitting, larger models are not necessarily better, so the challenge is to select a model of intermediate size that is best at what is really important, out-of-sample prediction. To do this, one must use different samples of the data for different parts of the procedure. Ideally, one portion of the data (a training set) is used for fitting, another portion (a *validation set*) for model selection, and a third portion (a *test set*) for final evaluation of predictive performance ([9], p. 222).

A final important consideration is the particular measure used for evaluating classifier performance. Any item processed by a binary classifier falls into one of four groups, defined by its true class (0 or 1) and its predicted class (0 or 1). The rates of these four outcomes can be displayed in a so-called confusion matrix, as shown in Table 1. The values $a,\ b,\ c,\ d$ in the table are the rates (relative frequencies) of the four possible outcomes. They must sum to 1. The values b and c (shown in bold) are the rates of the two types of errors: nonsmoke classified as smoke, and smoke classified as nonsmoke. The row sums f_0 and f_1 are the true proportions of items in each class.

Three error rates derived from the confusion matrix are considered subsequently. The *overall error rate* (OER$=b+c$) is simply the global proportion of pixels misclassified. The *classwise error rates* are the rates of misclassification in each class considered separately. We denote these by CER0$=b/f_0$ for the nonsmoke class, and CER1$=c/f_1$ for the smoke class.

Minimizing the OER will be taken as the primary goal of classifier construction. Note however, that our data set consists of 90% nonsmoke pixels (f_0=0.9), so focusing on overall prediction performance implicitly puts more weight on prediction accuracy in the nonsmoke class. Because the data are so unbalanced, even the naïve classification rule "assign all pixels to class 0" can achieve an error rate of only 10% (OER=0.1), but with the highly unsatisfactory classwise rates CER0=0 and CER1=1. More will be said about the trade-off between OER and CER in later discussion.

		Prediction		
		Class 0	**Class 1**	**Sum**
Truth	Class 0	a	$\mathbf{b}$	$a+b=f_0$
	Class 1	$\mathbf{c}$	d	$c+d=f_1$
	Sum	$a+c$	$b+d$	1

Table 1. A confusion matrix. Values in bold represent errors.

3. Experimental methods

The methods just described were applied to the full set of hyperspectral data. The logistic regression classifier was used, just as in the example. In the full-scale analysis, however, it was necessary to handle a much larger data set and a much larger pool of predictor variables. The following sections describe the methods used for preparing the data and searching for a suitable classifier.

3.1. Data splitting and sampling

This analysis took place in a data-rich context. Having a high volume of data is very advantageous, since the available pixels can be split into separate training, validation, and test groups with each group still having more than enough pixels to yield good estimates of the various quantities of interest. The data were randomly split into these three groups at the image level, with a roughly 50/25/25% split: 70 images (82×10^6 pixels) for training, 36 images (42×10^6 pixels) for validation, and 37 images (43×10^6 pixels) for testing.

The drawback of having this much data is the level of computational resources required to handle it. Fitting the logistic regression model requires matrix computations that are memory and computation intensive when the number of cases (pixels) or the number of predictors become large. To estimate a model with the 35 spectral bands as predictors using the full set of training images, for example, approximately 23 GB of RAM is be required just to hold the data in memory. Special techniques are required to perform regression computations on data sets this large. Furthermore, it is necessary to perform model fitting iteratively as part of a model search step, so simple feasibility is not sufficient. Computational run time is also an important factor.

A practical approach to working with such large data sets is to randomly sample a manageable subset of the data, and work with the sample instead. This approach will work well if the sample size can be chosen such that the computations are feasible and sufficiently fast, while still providing estimates of needed quantities (coefficient estimates, prediction error rates) that are sufficiently accurate.

To determine whether such a sample size could be found in the present case, a sequence of preliminary trials was carried out on the test and validation images. In these trials, the model with 35 main effects was fit to numerous independent training samples, and predictions were made on numerous independent validation samples. It was found that sampling 10^5 pixels was adequate for both the training and validation data. At this sample size, predicted probabilities from fitted models exhibited only minor variations (typically differing less than 0.02) when computed from different samples. Similarly, when the validation sample was this size, estimates of prediction error had variance low enough that it should be possible to estimate the prediction error rate on the full validation set to better than the nearest percentage point.

A working sample of 10^5 pixels was therefore drawn from the test images, and an equal-sized sample was drawn from the validation images. Subsequently all parameter estimation and model selection was done using these two samples, rather than the original images.

3.2. Model families considered

In an attempt to build a successful classifier, four groups of models were considered. Each group was defined by i) the set of candidate predictors that have the opportunity to be selected in the model, and ii) the methods used for model selection and model fitting. We attempted to find a single "best" classifier within each group, and carried forward those four best models for subsequent performance evaluations.

Scenario 1: *RGB model*. This model was the same as the first classifier shown in the earlier example, except with all three variables (R, G, B) used instead of only two. This model was included only as a reference point, since it was not expected to perform particularly well. There is only one possible model in this group, so no model selection step was necessary. Coefficients were estimated in the usual least-squares manner for logistic regression.

Scenario 2: *main effects model*. This model family used the 35 hyperspectral bands as candidate predictors. An optimal model with 35 or fewer variables was to be chosen by subset selection. Coefficients were estimated by least squares.

Scenario 3: *all effects model (subset selection)*. The third set of models included a greatly expanded set of predictors. The complete set of candidate variables for this case includes the following sets of variables:

- All 35 main effects.
- The 35 square-root terms.
- The 35 squared terms.
- The 595 interactions between different main effects.
- The 595 interactions between different square-root terms.
- The 595 interactions between different squared terms.
- The 1225 interactions between main effects and square-root terms.
- The 1225 interactions between main effects and squared terms.

In all, there are 4340 candidate variables in this collection. A best model consisting of a (relatively) small portion of these variables was found by subset selection, and coefficient estimation was done by least squares.

Scenario 4: *all effects model (LASSO selection)*. The fourth group of models used the same set of 4340 candidate predictors, but with model selection and parameter estimation carried out using the LASSO technique. Briefly, LASSO is a so-called *shrinkage* or *regularization* method, where parameter estimation and variable selection are done simultaneously. It works by introducing a penalty term into the least squares objective function used to fit the model. The nature of the penalty is such that certain coefficients are forced to take the value zero, effectively eliminating the corresponding variables from the model. The size of the penalty is controlled by a parameter; the larger this parameter, the more variables are removed from the model. The reader is referred to the literature for further details on LASSO and other shrinkage methods (for example, [11, 12, 9]). The LASSO-regularized logistic regression classifier was constructed using the R package glmnet [13].

3.3. Model selection

The main effects and all effects models required model selection by *best subsets*. For a given set of candidate predictors, this approach to model selection depends on two things: an objective function defining how “good” a particular model is, and a search procedure for finding the best model among all possibilities.

In the present case we were interested in out-of-sample prediction performance, so we used the validation sample of pixels to measure the quality of any proposed model. A straightforward measure of model quality is the prediction error rate on the validation data. While this measure could have been used, here a quantity known as *deviance* was used instead. The deviance is defined as −2 times the log-likelihood of the data under the model, and can be interpreted as a measure of lack of fit (smaller deviance indicates a better fit). For the logistic regression model with n pixels, the deviance is

$$-2\sum_{i=1}^{n} d_i, \quad \text{where} \quad d_i = \begin{cases} \log\left(\hat{\pi}_i\right) \text{ if pixel } i \text{ is smoke} \\ \log\left(1-\hat{\pi}_i\right) \text{ if pixel } i \text{ is nonsmoke,} \end{cases} \tag{3}$$

where $\hat{\pi}_i$ is the predicted probability of pixel i being in class 1. We can see from the equation that the i th pixel's deviance contribution, d_i, shrinks to zero when the predicted probability gets closer to the truth (i.e., when a smoke pixel's predicted probability approaches one, or when a nonsmoke pixel's predicted probability approaches zero). An advantage of the deviance is that it depends in a smooth and continuous way on the fitted probabilities, whereas the prediction error depends only on whether the $\hat{\pi}_i$ values are greater or less than the cutoff c.

In best subsets search, then, the objective function value for any proposed model was found by first estimating the model's coefficients using the training data, and then computing the deviance of the fitted model on the validation data.

Having defined an objective function, it was necessary to search through all possible models to find the best (i.e., minimum deviance) one. This task is challenging, because the combinatorial nature of subset selection causes the number of possible models to grow very quickly when the number of candidate predictors becomes large.

Let the *size* of a particular model be the number of predictors in the model, not including the intercept. Denote model size by k. For the main effects scenario with 35 predictors, there are a manageable 6454 possible models when $k=3$ (i.e., there are 6454 combinations of 3 taken from 35). When $k=5$, however, there are about 325 thousand models from which to choose; and when $k=15$, there are 3.2 billion models. For the all effects scenario with 4340 predictors, the situation is naturally much worse. Even for models of size 3, there are about 13.6 billion possible choices. For larger values of k, the number of possible models becomes truly astronomical, with approximately 10^{30} ten-variable models and about 10^{154} 70-variable models.

Clearly, it is not feasible to search exhaustively through all possible models for either the main effects or all effects scenario. Rather, a search heuristic is required to find a good solution in reasonable time. A traditional approach in such cases is to use sequential model-building procedures like forward, backward, or stepwise selection [14]. These methods have the advantage of convenience, but they lack a valid statistical basis and are generally outperformed by more modern alternatives.

An alternative option, that was pursued here, is to use a more advanced search heuristic to search the space of possible models. We used the function kofnGA, from the R package of the same name [15], to conduct model search using a genetic algorithm (GA). This function searches for best subsets of a specified size, using a user-specified objective function (which we chose to be the validation-set deviance). Instead of considering all possible model sizes, separate searches were run at a range of chosen k values. These were:

For the main effects model: $k=3, \ 5, \ 10, \ 15, \ 20, \ 25, \ 30.$

For the all effects model: $k=3, \ 10, \ 20, \ 30, \ 40, \ 50, \ 60, \ 70.$

By running the search at only these sizes, we expected to find a model close to the optimal size, without requiring excessive computation times. A discussion of GA methods is beyond the scope of this work, but references such as [16, 17, 18, 19] can be consulted for further information.

When using a search heuristic like GA on a large problem like this, we do not expect that the search will result in finding the single globally-optimal model in the candidate set. In fact if we were to run the search multiple times, it is likely that a variety of solutions will be returned. Nevertheless, the GA can be expected to find a good solution—that is, one with a validation-set deviance close to the minimum—in reasonable time. In practice we expect any model near the minimum deviance will have nearly equivalent predictive performance.

The model selection in the LASSO scenario was done quite differently. As mentioned previously, the LASSO solution depends on a regularization parameter that controls the complexity of the fitted model. For any given value of this parameter, a single model results, with some coefficients zero and some nonzero—the size of the model is implicit in the solution, and is not directly controlled. Model selection thus involves choosing only the value of the regularization parameter. Following the advice of [13], we used validation-set deviance as the measure of model quality for the LASSO fit, and chose the regularization parameter to minimize this quantity.

Note that the LASSO approach enjoys a computational efficiency advantage over the GA-based subset selection approach. For our large training and validation samples (10^5 pixels), fitting the LASSO at 100 values of the regularization parameter took approximately two hours on a contemporary desktop system, while a the longer GA runs (say, with all effects and $k = 50$) took an entire day. Given the overall timeframe of a study like this one, however, the run time difference is not viewed as especially important.

3.4. Performance evaluation

Predictive performance of the best models selected from each group was measured by the overall and classwise error rates OER, CER0, and CER1, as defined in Section 2.2. The probability cutoff c used to map the fitted probabilities onto the two classes was set to its default value of 0.5 for this performance comparison. There is no guarantee that 0.5 actually provides the best value, however. To investigate the impact of varying c, performance of the best model in group 3 was evaluated at a range of c values.

As an adjunct to quantitative assessment, qualitative analysis of model predictions was carried out by visual inspection of the predicted probability maps—greyscale images in which the intensity range [0, 1] represents the predicted probability of each pixel being smoke—from the best model in group 3. For all 37 test images, the probability maps were compared to the original RGB images, to learn more about which aspects of smoke detection were done well, and which were done poorly.

4. Results

The data splitting, sampling, and model selection procedures just described were carried out on the study data, with the net result of producing one best classifier from each of the four scenarios. These four best classifiers were subsequently used to generate predictions for every pixel in the 37 test images. The results of these tasks are presented below, beginning with model selection, and then moving on to the quantitative assessment of prediction performance. The qualitative assessment of performance is reviewed in Section 5.

4.1. Model selection results

The results of model selection are shown in Table 2 and Table 3. The first table lists all of the models considered, along with their deviance and their error rates on the validation data. The error rate estimates in the table are preliminary only, because they are measured on the same validation sample that was used to do variable selection. The final and most accurate measure of out-of-sample predictive performance (the error rates on the test images) are reported in the next section.

The four models selected as best in the four groups are shown in bold in Table 2. For model 1 (RGB), there was only one model, which was selected best by default. For models 2 and 3 (the main effects and all effects models), the best models had $k=20$ and $k=50$, respectively. For model 4 (the LASSO), the minimum-deviance approach chose a model with 109 variables.

Scenario/Model	**Results on VALIDATION sample**			
	Deviance	**OER (%)**	**CER0 (%)**	**CER1 (%)**
1. RGB	**58549**	**10.2**	**4.3**	**98.4**
2. Main effects, k variables				
$k=3$	57245	10.0	0.0	100.0
$k=5$	53162	9.8	0.4	94.3
$k=10$	50399	9.3	0.5	87.9
$k=15$	48521	8.7	0.5	83.0
***k*=20**	**48483**	**8.6**	**0.5**	**82.1**
$k=25$	48704	8.8	0.6	82.8
$k=30$	50144	8.8	0.7	81.7
3. All effects, k variables				
$k=3$	51262	9.6	0.4	92.9
$k=10$	42442	7.6	1.1	65.9
$k=20$	40180	7.2	1.1	62.2
$k=30$	39785	7.1	1.2	60.0
$k=40$	38600	6.8	1.3	55.7
***k*=50**	**38174**	**6.8**	**1.4**	**56.0**
$k=60$	38424	6.9	1.6	54.5
$k=70$	38475	6.8	1.6	53.7
4. All effects, LASSO*	**47711**	**8.1**	**1.6**	**66.6**

*The LASSO model shown is the minimum-deviance one, which had 109 nonzero coefficients.

Table 2. List of models considered, with results for the validation set.

Table 3 shows the particular combinations of variables that were chosen in the best models from each of the four groups. The main-effects-only model had 20 variables, the all-effects model had 50 variables, and the LASSO model had 109 variables (of which only 50 are shown). When regression models become this large, it is very difficult to glean any useful information from lists of included variables. Nevertheless, the table is presented for the sake of completeness.

1. RGB Image:
Red, Green, Blue (nonlinear transformations of bands 1, 4, and 3)
2. Main effects, *k*=20:
21, 31, 32, 24, 25, 36, 18, 7, 1, 23, 17, 6, 8, 30, 13, 11, 14, 16, 26, 15
3. All effects, *k*=50:
$\overline{19}:\overline{21}$, $\overline{26}:\overline{21}$, $24:\overline{26}$, $\bar{3}:\overline{28}$, $\bar{5}:\overline{24}$, $23:\bar{1}$, $28:\bar{4}$, $\bar{2}:\overline{25}$, $33:\overline{21}$, $\overline{30}:\bar{7}$, $\bar{8}:\overline{25}$, $27:\overline{31}$, $30:\underline{20}$, $22:\bar{2}$, $8:23$, $31:\overline{30}$, $\bar{9}:\overline{18}$, $24:\overline{27}$, $\underline{23}:\underline{8}$, $30:\underline{34}$, $27:\overline{23}$, $8:\underline{32}$, $11:\overline{36}$, $11:36$, $\bar{4}:\bar{9}$, $16:\underline{19}$, $23:\overline{19}$, $27:\overline{36}$, $7:\underline{23}$, $11:\underline{6}$, $\overline{16}:17$, $5:\underline{1}$, $20:\underline{14}$, $\underline{19}:\underline{25}$, $\overline{36}:\overline{35}$, $\underline{7}$, $\overline{16}:\overline{17}$, $20:\overline{34}$, $22:\underline{5}$, $\overline{23}:\overline{25}$, $\underline{12}:\underline{26}$, $\underline{13}:33$, $\underline{7}:27$, $19:\underline{13}$, $\bar{8}:\bar{6}$, $14:\underline{16}$, $35:\underline{8}$, $23:\bar{8}$, $11:\overline{16}$, $35:\overline{12}$
4. All effects, LASSO (109 variables, first 50 shown):
$\overline{24}:\overline{26}$, $\overline{20}:\overline{26}$, $\overline{18}:\overline{25}$, $\bar{7}:\overline{24}$, $6:\overline{24}$, $\overline{22}:\overline{26}$, $1:\overline{30}$, $1:\overline{25}$, $18:\overline{25}$, $4:\overline{23}$, $1:\overline{32}$, $32:\underline{7}$, $8:\overline{32}$, $23:\underline{4}$, $13:\overline{31}$, $3:\bar{3}$, $31:\underline{26}$, $2:\overline{30}$, $\overline{22}:\overline{36}$, $31:\underline{18}$, $\overline{10}:\overline{25}$, $\overline{24}:\overline{27}$, $26:\overline{22}$, $26:\overline{20}$, $17:\overline{22}$, $20:\overline{36}$, $\overline{10}:\overline{32}$, $31:\underline{10}$, $\overline{16}:\overline{31}$, $\overline{27}:\overline{31}$ $30:\underline{10}$, $23:\underline{9}$, $4:\overline{36}$, $\bar{5}:\overline{27}$, $\bar{6}:\overline{18}$, $\bar{9}:\overline{32}$, $7:\overline{18}$, $\bar{4}:\overline{36}$, $\overline{11}:\overline{20}$, $4:7$, $13:\overline{24}$, $21:\underline{27}$, $\overline{31}:\overline{32}$, $31:\overline{31}$, $31:\underline{4}$, $\overline{27}:\overline{33}$, $\bar{3}:\bar{9}$, $16:\overline{31}$, $26:\overline{27}$, $\overline{20}:\overline{23}$

Table 3. Chosen variables for the best model in each category. Variables are listed in descending order of coefficient magnitude. See the text for a description of the notation.

A compact notation is used in the table to reduce the space consumed by long lists of variables. In this notation, each of the 35 spectral bands in the original images (the main effects) is represented by its band number. Squared terms are written with a bar over the band number, and square root terms are written with a bar underneath. Interactions between two terms are indicated by a colon. So, for example, the notation $\underline{9}$ refers to the square root of band 9, and $11:\overline{17}$ refers to the interaction between band 11 and the square of band 17.

4.2. Predictive performance

The final estimate of the performance of the four selected models is based on those models' predictions on the complete set 37 test images. Together these images contain over 43 million pixels that were not used in any way during the model fitting and variable selection processes.

Because they are previously unused, they provide a more accurate approximation of the predictive power of the models (better than the validation data, which was not used for parameter estimation, but was used repeatedly for variable selection). The results are shown in Table 4.

	OER (%)	CER0 (%)	CER1 (%)
Model 1: RGB image	10.4	0.5	98.6
Model 2: main effects, 20 variables	8.6	0.5	82.1
Model 3: all effects, 50 variables	8.1	1.9	63.5
Model 4: all effects, LASSO (109 variables)	7.8	1.2	66.0

Table 4. Summary of the selected models and their predictive performance on the test images.

Figure 4 illustrates the trade-off between the different error types as the cutoff *c* is varied, for the 50-variable all effects model. The plot shows OER, CER0, and CER1 as functions of the cutoff. We can see that the overall error rate is in fact minimized at the original cutoff of 0.5, so changing the cutoff to improve performance on the smoke class will unfortunately come at the cost of worse overall performance. This notwithstanding, both OER and CER0 are relatively flat over the cutoff range (0.3, 0.5). So, for example, setting the cutoff to 0.4 will reduce the classwise error rate of smoke pixels to 50%, while increasing the OER only slightly.

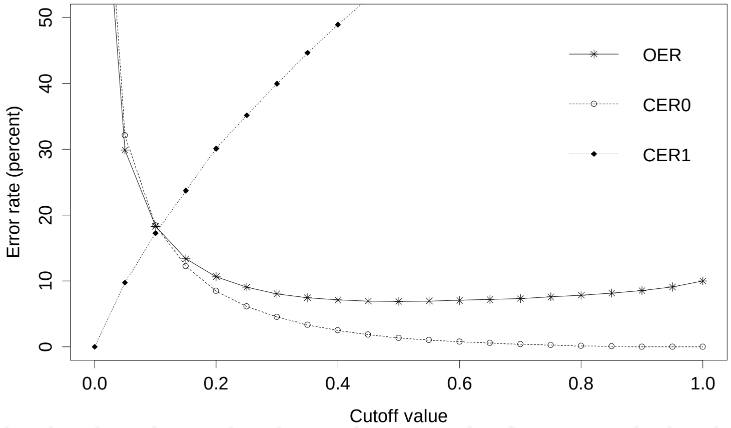

Figure 4. The effect of the decision cutoff on the overall and classwise error rates, for model 3.

5. Discussion

The experimental results are interpreted and discussed below, beginning with several remarks about model selection and performance evaluation, and followed by a qualitative evaluation of the classification results. Afterwards, a variety of suggestions for further improvement are provided.

5.1. Remarks on the selected models

The classification error rates were reported in Table 2 (for all models, on the validation set) and Table 4 (for the best models in each group, on the test set). Considering these tables, we see that our concern about the dominance of the smoke class (class 0) in the data set was justified. All of the models had overall error rates less than about 10%, which seems good at first glance. However in all cases this low error rate was achieved by having a very low error rate in the nonsmoke class (CER0) and a high error rate in the smoke class (CER1). This problem is particularly severe for smaller models and smaller sets of candidate variables, but even the best model in group 3 (the 50-variable model) had 56% misclassification of the smoke pixels.

Comparing the best models from each group, the only two models that can be considered even moderately successful are the two largest ones, the 50-variable all effects model (model 3) and the 109 variable LASSO model (model 4). There is little to separate these two classifiers: both have overall error rates of about 8% on the test set, with model 4 having a slight advantage; but model 3 has better performance on the smoke class.

Interestingly, these two models share only one variable in common (it happens to be $11:\underline{6}$). This is a consequence of the huge feature space and of the correlations among predictors. Two different models containing disjoint sets of variables can both have similar predictive power. This observation is related to the following two remarks.

Remark 1: physical interpretability of selected variables. It is desirable from a scientific and intellectual standpoint to be able to interpret the structure of a predictive model in terms of physical principles, but this is not always straightforward in a machine learning context. In the case of the spectral signature of smoke, a few general characteristics have been observed. Smoke scatters visible light [20], a component of it (organic carbon) is strongly absorbing below about 0.6 μm [21], and it is largely transparent in the middle infrared [22, 23]. We endeavored to interpret our models in light of these observations, but were unable to find any simple and unambiguous relationships based on the patterns of variables included in the models. This is often the price to pay for focusing on out-of-sample predictive accuracy: the classifier becomes a "black box" with internal structure that defies simple interpretation.

Remark 2: interpretability of model coefficients. Noticeably absent from the discussion so far has been the actual values of the regression coefficients in the fitted models. This has been deliberate, because in a pure classification problem like this one the predictive performance of the model as a whole is the overriding concern. Interpretability of model coefficients is desirable, but is likely not achievable when we have models with dozens of predictors that are all interactions. Assessment of statistical significance of particular predictors also adds nothing to our understanding of the model as a classifier, and is best avoided.

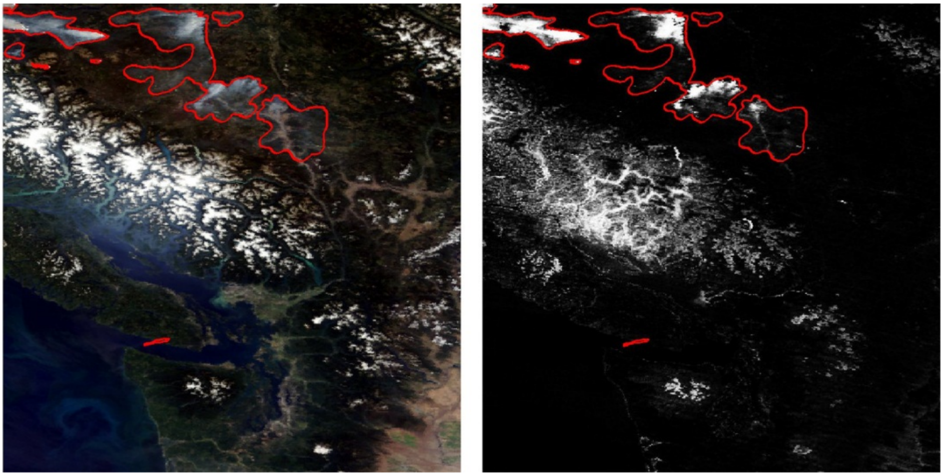

Figure 5. Results on a test image. Left: the RGB image. Right: the predicted probability map using the 50-variable model. The red contour delineates the true smoke region.

5.2. Qualitative performance analysis

Based purely on the observed numerical measures of prediction accuracy, it seems clear that none of the classifiers considered have performance good enough for real-world application, primarily because the majority of smoke pixels are misclassified in all cases. Visual inspection of the predictions on the test images can yield further insight into the nature of the problem, and possible causes of difficulty. Figure 5 and Figure 6 provide prototypical examples drawn from the test images. Our qualitative conclusions about predictive performance, based on the full set of 37 images, are listed below.

1. *Smoke-free images are generally classified well.* The classifier does have *some* ability to detect smoke, so it is still encouraging to observe that smoke-free images, or large regions that are smoke-free, are generally classified accurately. This can be observed in the bottom and left portions of Figure 6, which are assigned low probabilities throughout, despite the presence of clouds, water, and various types of terrain.

2. *Clouds and smoke can be distinguished well from one another.* It was observed that throughout the 37 test images, there were very few instances where cloud was erroneously identified as smoke. This provides at least some encouragement that the use of hyperspectral data holds benefits, because distinguishing clouds from smoke visually using the RGB images can be quite difficult.

3. *Snow and ice can be distinguished from smoke, but with greater difficulty.* A similar comment can be made about snow and ice, but less emphatically. The classifier generally performed well in separating smoke from snow and ice, but performance was less consistent. In certain images this task seemed to pose no problem, while in other images significant

numbers of snow or ice pixels were incorrectly labelled smoke. Both Figure 5 and Figure 6 provide some evidence of this, with moderate probabilities being mapped over the Coast Mountains in the upper left of either image.

4. *Co-located smoke and clouds present a problem.* The starting point for this problem is the assumption that smoke and clouds may both exist in the same pixel. Separation of smoke from clouds when both are in the same vicinity is a problem in two respects. First, when the masks were being prepared it was extremely difficult for the human interpreter to decide whether or not a given pixel in a cloudy region actually contains smoke. When clouds and smoke are mixed or adjacent, it is very difficult to distinguish one from the other using the RGB image alone. Second, because cloud is a significant constituent of the nonsmoke pixel class, the classifiers learned to assign low probability to pixels with the characteristics of clouds. An example of this problem can be seen in the upper right corner of Figure 6. In the RGB image, it is unclear if the bright feature in this corner is a cloud, and if so, whether there is also smoke present. From the probability map, it appears that there was indeed cloud in this region, which caused it to be assigned low probability.

5. *Prediction maps are unrealistically noisy.* Our mental model of the true scene in these images is of smoke regions being contiguous with relatively smooth boundaries. Because we are classifying pixels independently, however, this information is not incorporated into our procedures. The noisy nature of the probability maps is visible in both the smoke and nonsmoke regions ofFigure 5 andFigure 6.

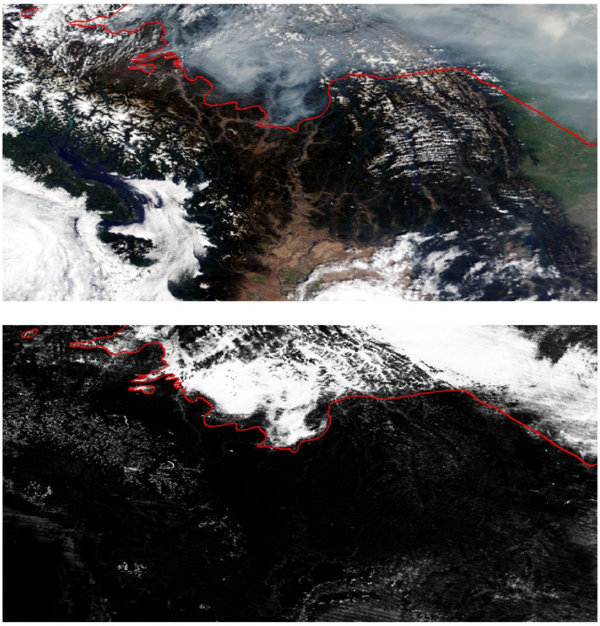

Figure 6. Another example prediction. Top: RGB image. Bottom: predicted probability map.

6. *The quality of the training data is a major impediment to classifier construction.* Perhaps the most significant problem inherent in this study is uncertainty about the assigned classes in the original images themselves. Various portions of the images proved extremely difficult to assign to one class or the other with high confidence during the masking step. The aforementioned regions of mixed smoke and cloud provide one example. Regions where smoke becomes less concentrated provide another example (see Figure 5): where does the smoke end and the nonsmoke begin? In the same figure, we see a third example. A large number of pixels in a region over the mountains are "erroneously" assigned a high probability of being smoke. Is this a classification error, or an error in masking the original RGB image? The RGB image has a hazy appearance in this region, but it was not assigned to the smoke class due to the absence of a local fire and the general uncertainty about the nature of this hazy appearance. After the fact, it seems plausible that the classifier is detecting smoke that was erroneously labelled nonsmoke in the data set.

5.3. Opportunities for improvement

While the classification results were mixed, we feel there were enough positive elements to warrant further investigation, and that the overall approach can still be successful with appropriate modifications and extensions.

Probably the clearest opportunity for improvement is to alleviate the uncertainty in the true class labels that exists throughout the data set, and was illustrated in Figure 5 and Figure 6. The ambiguity in distinguishing smoke from nonsmoke at various places in the RGB images is a fundamental limitation. Simple approaches to solving this problem include considering only smoke plumes or "thick" smoke; excluding pixels that the photointerpreter finds ambiguous or that contain both cloud and smoke; or labelling images with more than two classes. More involved approaches include modelling each pixel as a mixture of different components, or modelling some continuous measure of smoke concentration rather than a binary presence/absence response. An unsupervised learning (clustering) approach or a semi-supervised method (where only some pixels are labelled) could also be considered, though such methods make quantitative performance assessment more difficult.

Another avenue for potential improvement of classification performance is to modify the feature space in the logistic model in the hopes of improving the separability of smoke and nonsmoke. While this could be done by adding even more factorial terms (cubic terms, higher-order interactions, and so on), it is unlikely that the benefit of doing so would outweigh the increase in computational burden. Instead, more focused modifications of the model could be considered. To reduce the effect of highly heterogeneous surface terrain in the nonsmoke class, for instance, a baseline spectrum (perhaps taken as an average of observations over recent clear-sky days) could be included as predictors in the model. Or each pixel could be assigned to a known ground-cover class at the outset, and these classes could be included in the model as categorical variables. Another option is to replace the fixed powers of reflectance we used (squared and square root terms) with spline functions, allowing data-adaptive nonlinear transformations of the variables to be used in the model. We anticipate exploring some of these alternatives in future work with these data.

Additional possibilities for improvement can be found by moving farther from the logistic regression framework. Under the assumption of independent pixels, for example, any of the many existing classification tools could be applied to the data. The support vector machine (see, e.g., [24], Ch. 11) in particular is a state-of-the-art method that has performed well across a variety of tasks and is worthy of consideration. If the independence assumption is dropped, the autologistic regression model [25], a model for spatially-correlated binary responses, is a natural fit for these data. This model would alleviate the problem of noise in the predicted probabilities, producing smoother and more accurate prediction maps. It is a natural extension of logistic regression to spatially-associated data. Finally, it may also be possible to incorporate relevant ancillary information (for example, prior knowledge of fire locations and wind directions) into a classification model to improve predictive power. Again, consideration of these alternatives and extensions are planned in future work.

6. Conclusion

The smoke identification problem provided a case study on the use of supervised learning to automate the process of recognizing features of interest in remote sensing images. The machine learning approach is especially attractive when working with hyperspectral images, because the high dimensionality of the data makes it very challenging for a human photointerpreter to consider all of the potential relationships in the data. Subject-matter knowledge can help to focus a human expert on certain models, relationships, or spectral bands, but automated procedures provide a valuable complementary approach. They can be used to search for more complex or previously unconsidered relationships, driven by the data itself. If a machine learning procedure can be implemented successfully, another clear benefit is the ability to process data at a speed and scope not feasible by other means.

Our primary conclusion regarding the smoke identification goal is that the spectral information in the smoke and nonsmoke classes overlap to such a degree that it is not possible to construct a highly successful classifier—at least with the models and methods we employed. The results have some promising elements, however. Notably, it appears possible to distinguish smoke from cloud and snow when a) the smoke is not mixed with cloud, and b) the smoke is not too diffuse. Indeed, if the goal of the study were to find clear-sky smoke plumes only, the approach would be quite successful. Classification errors were largely attributable to the presence of cloud in a smoky region, to the smoke being too diffuse, or to inaccuracies introduced in the initial labelling of the data. Armed with this understanding, it should be possible to make considerable improvements to the results with adjustments to the methodology.

The problem used for this case study is a challenging image segmentation task, made more challenging by the loose definition of “smoke” used in the initial labelling of the data set. Reflecting this, the best classifiers we found were only partially successful. Still, the process of developing them has helped to provide insight into the problem and allows us to present both the advantages and challenges of the machine learning approach. With the dimensionality and throughput of remote sensing data ever on the rise, computer intensive techniques such as those explored here will be of increasing importance in the future.

Author details

Mark A. Wolters[1*] and C.B. Dean[2]

*Address all correspondence to: mwolters@fudan.edu.cn

1 Shanghai Center for Mathematical Sciences, Fudan University, Shanghai, China

2 Department of Statistical and Actuarial Sciences, Western University, London, Canada

References

[1] Richards JA. Remote Sensing Digital Image Analysis. 5th ed.: Springer-Verlag; 2013.

[2] R Core Team. R: A Language and Environment for Statistical Computing Vienna: R Foundation for Statistical Computing; 2014.

[3] National Aeronautics and Space Administration. MODIS Web. [Online].; 2014 [cited 2014 November 25. Available from: http://modis.gsfc.nasa.gov/about/specifications.php.

[4] MODIS Characterization and Support Team. MODIS Level 1B Product User's Guide. Greenbelt, MD: NASA/Goddard Space Flight Center; 2012. Report No.: MCST Document # PUB-01-U-0202-REV D.

[5] Goddard Space Flight Center. LAADS Web. [Online].; 2014 [cited 2014 November 25. Available from: http://ladsweb.nascom.nasa.gov/.

[6] Gumley L, Descloitres J, Schmaltz J. Creating Reprojected True Color MODIS Images: A Tutorial. Maison, WI: University of Wisconsin-Madison, Space Science and Engineering Center; 2010. Report No.: Version 1.0.2.

[7] Giglio L, Descloitres J, Justice CO, Kaufman YJ. An Enhanced Contextual Fire Detection Algorithm for MODIS. Remote Sensing of Environment. 2003; 87: p. 273-282.

[8] Bishop CM. Pattern recognition and machine learning New York: Springer; 2006.

[9] Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed.: Springer Science+Business Media; 2009.

[10] James G, Witten D, Hastie T, Tibshirani R. An Introduction to Statistical Learning with Applications in R: Springer Science+Business Media; 2013.

[11] Tibshirani R. Regression Shrinkage and Selection via the Lasso. Journal of the Royal Statistical Society, Series B. 1996; 58(1): p. 267-288.

[12] Efron B, Hastie T, Johnstone I, Tibshirani R. Least Angle Regression. The Annals of Statistics. 2004; 32(2): p. 407-451.

[13] Friedman J, Hastie T, Tibshirani R. Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software. 2010; 33(1).

[14] Miller A. Subset Selection in Regression. 2nd ed.: Chapman & Hall; 2002.

[15] Wolters MA. A Genetic Algorithm for Selection of Fixed-Size Subsets, with Application to Design Problems. Journal of Statistical Software. (to appear).

[16] Michalewicz Z, Fogel DB. How to Solve it: Modern Heuristics. 2nd ed.: Springer-Verlag; 2004.

[17] Rothlauf F. Design of Modern Heuristics: Principles and Application Heidelberg: Springer; 2011.

[18] Gendreau M, Potvin JY, editors. Handbook of Metaheuristics. 2nd ed.: Springer; 2010.

[19] Whitley D, Beveridge JR, salcedo CG, Graves C. Messy Genetic Algorithms for Subset Feature Selection. 1997..

[20] Li Y, Vodacek A, Zhu Y. An automatic statistical segmentation algorithm for extraction of fire and smoke regions. Remote sensing of environment. 2007; 108(2): p. 171-178.

[21] Jethva H, Torres O. Satellite-Based Evidence of Wavelength-Dependent Aerosol Absorption in Biomass Burning Smoke Inferred from Ozone Monitoring Instrument. Atmospheric Chemistry and Physics. 2011; 11: p. 10541-10551.

[22] Miura T, Huete AR, van Leeuwen WJD, Didan K. Vegetation Detection Through Smoke-Filled AVIRIS Images: An Assessment Using MODIS Band Passes. Journal of Geophysical Research. 1998; 103(D24): p. 32001-32011.

[23] Chu DA, Kaufman YJ, Remer LA, Holben BN. Remote Sensing of Smoke from MODIS Airborne Simulator During the SCAR-B Experiment. Journal of Geophysical Research. 1998; 103(D24): p. 31979-31987.

[24] Izenman AJ. Modern Multivariate Statistical Techniques: Springer Science+Business Media; 2008.

[25] Hughes J, Haran M, Caragea PC. Autologistic models for binary data on a lattice. Environmetrics. 2011; 22(7): p. 857-871.

INTECH