Title

Variable Selection and Spatial Models for Identifying Chemical Plumes in Hyperspectral Imagery

Author

Mendoza, Nicole Andree

Publication Date

2018

UNIVERSITY OF CALIFORNIA
SANTA CRUZ

VARIABLE SELECTION AND SPATIAL MODELS FOR IDENTIFYING CHEMICAL PLUMES IN HYPERSPECTRAL IMAGERY

A dissertation submitted in partial satisfaction
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

STATISTICS AND APPLIED MATHEMATICS

by

Nicole Andree V. Mendoza

June 2018

This Dissertation of Nicole Andree V. Mendoza is approved:

Professor Abel Rodriguez

Professor Raquel Prado

Professor Juhee Lee

Tyrus Miller
Vice Provost and Dean of Graduate Studies

Contents

List of Figures

Abstract

Hyperspectral imaging is a remote sensing technique widely used in a variety of military and environmental applications. For example, hyperspectral images can be used to detect chemical plumes invisible to the human eye, and to identify their chemical structure. A hyperspectral image is a massive cube of data consisting of thousands of pixels each with dozens of observations over a range of frequencies in the electromagnetic spectrum. Algorithms that use hypothesis testing and assume independence over pixels have shown success in detecting gas clouds, but often fail in identifying chemical components. We approach identification problems in hyperspectral imaging as a variable selection problem, which can be solved robustly by taking advantage of spatial information in the image. For this purpose we develop Bayesian spatial model selection algorithms which use mixtures of g-priors, Gaussian Markov Random Fields, and Gaussian Process Priors to account for correlation among chemicals, to induce spatial dependence among pixels, and to account for nonlinearities in pixel signals. To illustrate the performance of the models we apply them to several partially synthesized hyperspectral images and show that our method outperforms state-of-the-art algorithms, such as the LASSO and Fused LASSO.

Acknowledgement

Thank you, Dimitris Manolakis, Steven Golowich, Eric Truslow, Robert DiPietro, and Michael Piper at MIT Lincoln for providing insights into the physics behind hypersepctral imagery and access to the embedding algorithm and raw hyperspectral images.

Next, I would like to recognize the SOE Graduate Advising team, especially Tracie Tucker and Lisa Slater, for all the administrative help they have provided during my time at UCSC

The work done for this dissertation would not have been possible without the SOE Tech Staff. Eric Shell and Heidi Sitton were instrumental in solving any computing issues I encountered with AMS's and SOE's Linux servers.

To my husband, Jeremy, and my parents, Janet and Emmy, thank you for all your efforts to assist me in any way possible so that I could fully focus on research.

For the camaraderie in coursework and extra curricular activities, many thanks to the students of AMS. I am forever grateful for Tatsiana Maskalevish and Tony Pourmohamad for being my study buddies during my first year in the AMS program. I am also deeply grateful for Sisi Song, Katelyn Byington, Annalisa Cadonna, and Brenda Betancourt for being my support system in Santa Cruz.

Tremendous praise goes to the entire AMS faculty for their expertise in their fields of research, and especially for their dedication to students as teachers and mentors. Many thanks to Professor Brummel for being on my advancement committee. Professor Lee, thank you for being on my defense committee. I will always be appreciative of you for helping me with my first academic conference talk, by listening to me practice it over, and over, again! Professor Prado, thank you for being part of my graduate student journey at practically every step of the way. From my first course (AMS 203), to being my Masters adviser, to being on my advancement committee, and now to being on my defense committee. Thank you for all this and for everything in between.

To my PhD adviser, Professor Rodriguez, I am forever indebted to you for the intellect, opportunities, and mentorship you have provided me over the years. The experience and exposure I gained through our work together, MIT Lincoln Labs internships, and the countless conferences near and abroad have been invaluable for my professional career, and it's all thanks to you. Lastly, I am truly grateful for your patience and perseverance in guiding me to complete my dissertation.

Chapter 1

Introduction

1.1 Remote sensing of gaseous plumes in hyperspectral imagery

A hyperspectral sensor simultaneously measures the distribution of radiation over hundreds of narrow and continuously spaced spectral bands in each pixel of an image (Schowengerdt, 2006). The resulting collection of images is organized as a“data cube” with pixels on the x- and y-axes, and electromagnetic radiation wavelengths on the z-axis (see Figure 1.1). Hyperspectral images often depict a natural scene captured from above, or from the side as shown in Figure 1.2. The hardware of hyperspectral sensors typically consists of an optical system, a scanning system, and a spectrometer. The optical and scanning systems take in light from the natural scene to produce the spatial dimensions of the image, and also control the image's pixel resolution. The spectrometer uses a spectral response function to produce the spectrum of each pixel and to control the image's spectral resolution. For example, the image in Figure 1.1 was produced using a “push broom” optical and scanning system with a prism-based spectrometer resulting in a 128x700 pixel image with 1.1 mrad resolution per pixel, and a spectral range in the longwave infrared (LWIR) region (8-13 μm) of the electromagnetic spectrum. For more on the mechanisms of hyperspectral sensors see Hackwell et al. (1996), Farley et al. (2006), and Manolakis et al. (2014).

Applications of hyperspectral sensing technology include environmental monitoring (Visser,

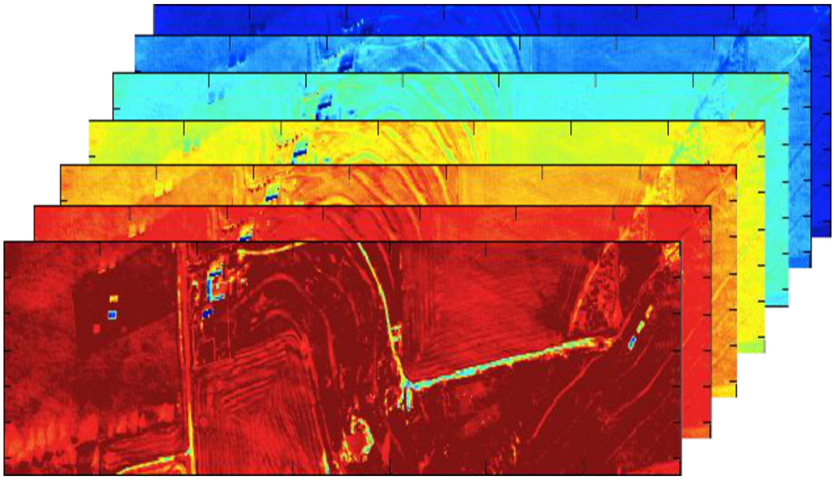


Figure 1.1: Slices of a downward looking hyperspectral image of a large field over wavelengths in LWIR. Captured by the Aerospace Corporation's Saptially Enhanced Broadband Array Spectograph System (SEABASS) hyperspectral imaging sensor at the Department of Energy's Atmospheric Radiation Measurement facility.

2000), agricultural monitoring (Thenkabail et al., 2012), food safety (Gowen et al., 2014; Lu and Chen, 1999), and chemical warfare (Demirev et al., 2005). Hyperspectral sensors can be used to remotely sense dangerous substances that are only observable in frequencies outside of the visible spectrum. Hence, a particularly important application of hyperspectral imagery is the remote sensing of gaseous plumes, which are usually undetectable to the naked eye. In this context, the literature usually distinguishes among three different tasks, namely, detection (which involves detecting if a gaseous plume is present), identification (which involves identifying exactly which chemicals make up the plume), and quantification (which involves measuring the amount of each chemical present in the plume). For a review, see Manolakis et al. (2014).

There is a well developed set of tools for detection of chemical plumes in hyperspectral imagery. Examples include geometric methods such as the Mahalanobis distance (Manolakis et al., 2001), and the generalized likelihood ratio test (GLRT), which is used widely and is referred to as the match filter detector in the signal processing literature (Kay, 1998). Messinger (2004) compares principal component analysis and projection pursuit methods to a spectral matched filter, and Theiler et al. (2005) develop non-linear matched filter methods for detecting plumes. Other variants of the matched filter are the adaptive match filter (AMF), and the adaptive coherence estimator, or ACE (Scharf and McWhorter, 1996). Theiler and Foy (2008) develop an elliptically-contoured GLRT detection method, and show that it has two limiting cases, which are AMF and ACE. Lastly, support vector machines have also been utilized for detection (Gurram and Kwon, 2010).

Methods for identification are comparatively less well developed, and Manolakis et al. (2014) show that methods based on the match filter often yield a high false positive rate when used for identification, i.e., they can detect that a plume is present, but they identify chemicals in the plume that are not actually part of its chemical structure. Many existing identification methods rely on tools for variable selection in linear models (e.g., Burr and Hengartner, 2006 and Bioucas-Dias et al., 2012). These include penalization methods such as the least absolute shrinkage and selection operator, or LASSO (Iordache et al., 2011; Tibshirani, 1996), and Bayesian variable selection techniques (Burr et al., 2008). However, the accuracy of these techniques is generally low, leading to a large number of false positives and false negatives. In particular, these techniques usually struggle to differentiate between chemicals with similar hyperspectral signatures.

For some applications fast algorithms for detection, identification, and quantification in hyperspectral imagery are necessary. For example, battlefield applications remotely sense dangerous substances in an area before a soldier enters the area. Such applications require algorithms that can produce results in real time (Chai et al., 1999), i.e., algorithms that need only seconds for processing time. Some methods discussed in this work are capable of near real time where processing time takes minutes. Other methods that we develop are well suited for precision agriculture applications, which detect diseases in crops (Mahlein, 2016) and predict crop growth and yield variability (Mahlein et al., 2012). Such applications do not require real-time analysis. Furthermore, forensic applications that detect chemicals in bioagents (Brewer et al., 2008) in hyperspectral data and artwork restoration applications that use hyperspectral imaging (Fischer and Kakoulli, 2006) also do not require real-time algorithms.

In this work we focus on developing supervised variable and model selection methodologies that can simultaneously perform detection and identification. In the next section we discuss the physical model of the signal entering the hyperspectral sensor that is the basis of our statistical models. Then in Sections 1.3 to 1.4 we build off the phyisical model to describe an embedding algorithm used to create partially synthesized datasets. In Section 1.5 we present the hyperspectral datasets that we will be analyzing in this manuscript, and in Section 1.6 we describe the metrics we will use to evaluate the detection and identification methods proposed in this work. In Chapter 2 we propose and evaluate supervised linear statistical methods that perform detection and identification under the assumption that all pixels in a hyperspectral image are independent. Next in Chapter 3 we again develop statistical methods for detection and identification that are supervised and linear, but we incorporate spatial information in these methods. We show that spatial methods outperform independent pixel-by-pixel methods. In Chapter 4 we build statistical models that are supervised and spatial, but now consider cases when nonlinearity in the signal must be accounted for. Then in Chapter 5 we discuss computational considerations for implementing all the methods. Lastly in Chapter 6 we close wtih conclusions and future work.

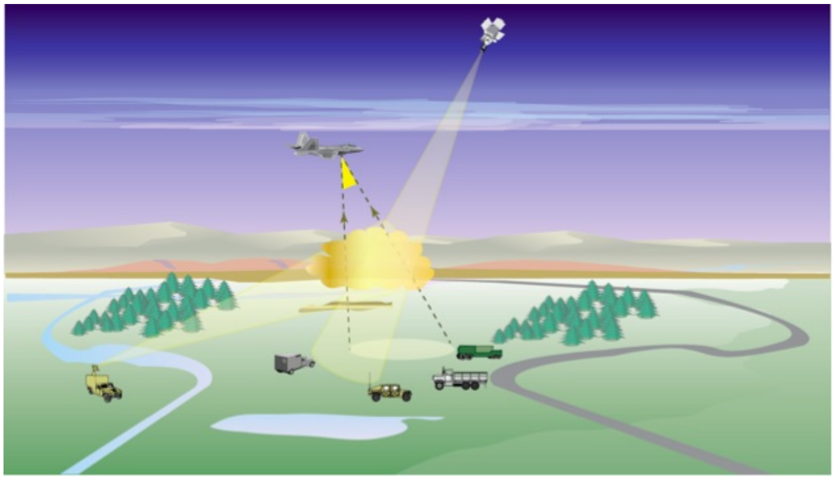


Figure 1.2: Hyperspectral sensors on planes or satellites can capture downward-looking images, while grounded automobiles can capture side-looking images. Here, the hyperspectral sensors are capturing a chemical plume within a natural scene.

1.2 Physical model

In this section we highlight key aspects of the physical model that underlies the hyperspectral sensor; for a detailed review please see Manolakis et al. (2014), which we follow closely. Please note that the discussion below focuses on a single pixel of the hyperspectral image and that we have dropped any subscript indexing the pixel to simplify our notation.

Radiative transfer theory allows us to model at-sensor radiance fairly simply given certain assumptions. Within the application at hand (remote sensing of chemical plumes), a hyperspectral sensor captures signals coming from three light sources: the background (such as a field, or building), the atmosphere, and the plume. We assume that the atmosphere and plume are free of aerosols and particulate matter, that the background, plume, and atmosphere have their own homogeneous temperatures, that the plume and atmosphere are homogeneous in composition, that the distance between the background and plume is small, and that the reflections off the background from the plume and down-welling atmospheric radiance are ignorable.

Using Kirchhoff's law (Thomas and Stamnes, 2002), a light source is a function of wavelength, λ, and can be modeled as,

$$L(\lambda) = [1 - \tau(\lambda)]B(\lambda, T), \tag{1.1}$$

where $\tau(\lambda)$ is a dimensionless ratio between 0 and 1 that describes the light source's transmittance, and

$$B(\lambda, T) = \frac{2hc^2}{\lambda^5} \frac{1}{\exp\left(\frac{hc}{\lambda kT}\right) - 1}, \tag{1.2}$$

is the Plank function, which describes radiation emitted by a blackbody at wavelength, λ, and temperature, T. The parameters, h, k, and c are the Plank constant, Boltzmann constant, and the speed of light respectively. Clearly, the physical model of any light source is non-linear, which can be a challenge for detection and identification algorithms.

In the sequel, let $L_p(\lambda)$ denote the plume signal at frequency, λ, $L_b(\lambda)$ be the signal from the background light, and $L_a(\lambda)$ be the signal from the atmosphere's light. The strength of the signal recorded at the sensor depends upon the transmittance of the plume, denoted by $\tau_p(\lambda)$, and that of the atmosphere, denoted by $\tau_a(\lambda)$ (see Figure 1.3). In particular, the at-sensor radiance signal when the plume is not present is given by

$$L_{\text{off}}(\lambda) = L_a(\lambda) + \tau_a(\lambda)L_b(\lambda), \tag{1.3}$$

and when the plume is present it is given by,

$$L_{\text{on}}(\lambda) = L_a(\lambda) + \tau_a(\lambda)L_p(\lambda) + \tau_a(\lambda)\tau_p(\lambda)L_b(\lambda). \tag{1.4}$$

A key observation is that in LWIR, the atmosphere has high transmittance i.e., $\tau_a(\lambda) \approx 1$.

Therefore the plume signal is primarily distorted by the background signal.

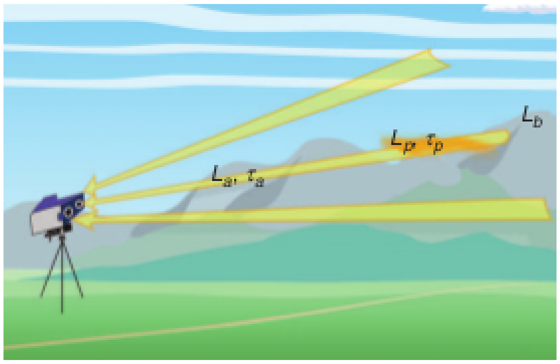


Figure 1.3: The hyperspectral sensor captures radiance from three layers, namely the atmosphere, plume, and background represented by L_a, L_p, and L_b, respectively. τ_a and τ_p represent the transmittance of the atmosphere and plume, respectively.

According to Beer's Law (Thomas and Stamnes, 2002) the transmittance of the plume can be expressed as

$$\tau_p(\lambda) = \exp\left(-\sum_{m=1}^{M} \gamma_m \alpha_m(\lambda)\right), \tag{1.5}$$

where M is the number of possible chemicals in the plume, $\gamma_m \geq 0$ is the concentration of chemical m in the plume, and $\alpha_m(\lambda)$ is the spectral signature of chemical m at frequency λ. Note that in LWIR many chemicals have unique spectral signatures, but are often correlated.

In summary, non-linearity in the light source and transmittance models, and the background contamination of the plume make detection a difficult task. Lastly, correlation among chemical spectral signatures poses additional challenges, especially for identification.

1.3 Embedding algorithm

Obtaining hyperspectral images with real plume releases is expensive, and difficult to obtain due to security clearance requirements. Although there are publicly available hyperspectral images with real chemical releases, the majority contain only a few plumes usually with only one chemical, and therefore are not useful for evaluating identification algorithms. Therefore we embed plumes with one or more chemicals on real plume-free hyperspectral images to create semi-synthetic datasets that are more useful for evaluating the detection and identification algorithms developed in this work.

In this section we describe the algorithm for embedding plumes in a hyperspectral image. We stress that embedding is a non-linear process that follows the physical model described in Section 1.2. Using the model of a light source in (1.1), and the Plank function in equation (1.2), we can rewrite equation (1.3) as,

$$L_{\text{off}}(\lambda) = (1 - \tau_a(\lambda))B(\lambda, T_a) + \tau_a(\lambda)L_b(\lambda), \tag{1.6}$$

while equation (1.4) can be expressed as,

$$L_{\text{on}}(\lambda) = (1 - \tau_a(\lambda))B(\lambda, T_a) + \tau_a(\lambda)\tau_p(\lambda)L_b(\lambda) + \tau_a(\lambda)(1 - \tau_p(\lambda))B(\lambda, T_p). \tag{1.7}$$

By adding and subtracting $(1 - \tau_a(\lambda))B(\lambda, T_a)\tau_p(\lambda)$ to the right-hand side of (1.7) we obtain,

$$\begin{split} L_{\text{on}}(\lambda) = \tau_p(\lambda)[\tau_a(\lambda)L_b(\lambda)(1 - \tau_a(\lambda))B(\lambda, T_a)] + \\ (1 - \tau_p(\lambda))[\tau_a(\lambda)B(\lambda, T_p) + (1 - \tau_a(\lambda))B(\lambda, T_a)]. \end{split} \tag{1.8}$$

Finally, by noting that the term in the first set of brackets on the right-hand side of equation (1.8) is exactly equal to $L_{\text{off}}(\lambda)$ we can simplify $L_{\text{on}}(\lambda)$ to,

$$L_{\text{on}}(\lambda) = \tau_p(\lambda)L_{\text{off}}(\lambda) + (1 - \tau_p(\lambda))[\tau_a(\lambda)B(\lambda, T_p) + (1 - \tau_a(\lambda))B(\lambda, T_a)], \tag{1.9}$$

which is the form of $L_{\text{on}}(\lambda)$ that is used for embedding.

1.4 Implementation of embedding algorithm

Matlab code for computing the embedding algorithm was provided to us by our collaborators at MIT Lincoln Laboratory. To implement the algorithm, first we need to deicide where to geographically embed plumes in the image. Then, all of the expressions and parameters on the right-hand side of equation (1.9) need to be estimated or specified by the user, except for $L_{\text{off}}(\lambda)$, which is provided to us by the reference hyperspectral image on which plumes are being embedded. Finally, we follow equation (1.9) to obtain the embedded signal.

1.4.1 Specifying the embedded plume's geographic location

In Figure 1.4 we show two pixel spectra associated with the plume free image shown in Figure 1.1. The pixel spectrum on the left is from a portion of the image where the background is composed of a field, whereas the pixel spectrum on the right is from a portion where the background is composed of buildings. We can see that the left pixel spectrum is relatively "flat" or smooth compared to the right pixel, which has sharp features. Plumes that are located on flat backgrounds will be easier to detect and identify than plumes located on non-flat backgrounds, which are noisier.

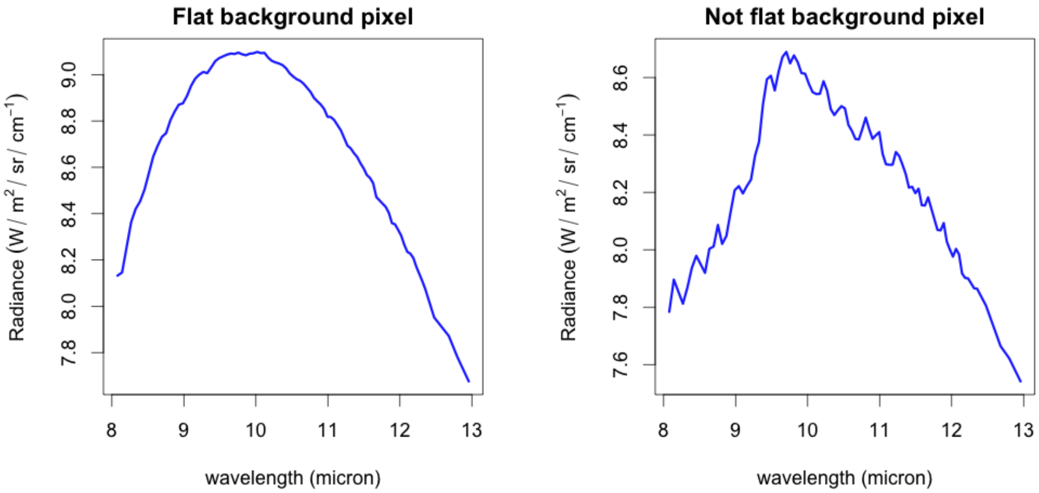


Figure 1.4: Left plot: Example of a background pixel spectrum that is flat. **Right plot**: Example of a background pixel spectrum that is not flat.

1.4.2 Estimating the atmosphere's transmittance, $\tau_a(\lambda)$, and temperature, T_a

We estimate the atmospheric parameters, $\tau_a(\lambda)$ and T_a, from the reference image on which the plume is being embedded. Estimating the transmittance of the atmosphere $\tau_a(\lambda)$ is a difficult problem (Manolakis et al., 2014). However, if the hyperspectral image has a downward-looking geometry (as is the case for the reference hyperspectral image used in our illustration), we can use the method developed by Young et al. (2002) to obtain an estimate for $\tau_a(\lambda)$. For estimating T_a, note that for the background hyperspectral image the minimum sampled wavelength was 8.0778 microns, and in this wavelength the atmosphere can be assumed opaque. This means that the radiance observed at this wavelength is well modeled by the Plank function in (1.2), and we can use its inverse to estimate the atmosphere's temperature in each pixel. The value of T_a is then set to be the median atmospheric temperature over all pixels.

1.4.3 Specifying the plume's transmittance, $\tau_p(\lambda)$, and temperature, T_p

The plume transmittance, $\tau_p(\lambda)$, is a tuning parameter that must be specified by the user. Recall that Beer's law establishes that the plume's transmittance can be expressed as,

$$\tau_p(\lambda) = \exp\left(-\sum_{m=1}^{M} \gamma_m \alpha_m(\lambda)\right).$$

The parameter, $\alpha_m(\lambda)$, is the spectral signature of the m-th chemical that the user wants to include in the embedded plume. The Pacific Northwest National Laboratory (PNNL) chemical spectral database (Sharpe et al., 2004) provides a library of hundreds of chemical spectral signatures in high spectral resolution. Note that the wavelengths at which these spectral signatures are captured do not necessarily match those associated with the background, so they typically need to be interpolated and down sampled. Because the original signatures have a very high-resolution, linear interpolation is used with very little loss of accuracy.

The parameter, γ_m, is the concentration of the m-th chemical present in the plume, and must be specified by the user. The values of γ_m control the dataset's level of difficulty. Since $\gamma_m = 0$ implies chemical m is not present in the plume, values of γ_m that are too close to 0 may be undetectable. Moderate values of γ_m also imply an optically thin plume whose transmittance can be modeled as approximately linear as opposed to the exponential form of Beer's law. Specifically, $\sum_m^M \gamma_m \alpha_m(\lambda) \ll 1$ when a plume is optically thin, and we can linearize Beers law using the first term of it's Taylor series expansion,

$$\tau_p(\lambda) \approx 1 - \sum_{m}^{M} \gamma_m \alpha_m(\lambda). \tag{1.10}$$

In Figure 1.5 we illustrate how well equation (1.10) (denoted by $\tilde{\tau}$) approximates Beer's law (denoted by τ), for different values of γ_m. Note that in the illustration we use the high-resolution spectral signature of chemical triethyl phosphate, or TEP, for $\alpha_m(\lambda)$. We can see that the approximation does relatively well for small of values of γ_m, like 1 and 5. However, as γ_m increases, and especially at $\gamma_m = 20$, the approximation is far away from Beers law. Therefore, optically thin plumes can be easier to detect and identify than opaque plumes when γ_m takes on a very large value.

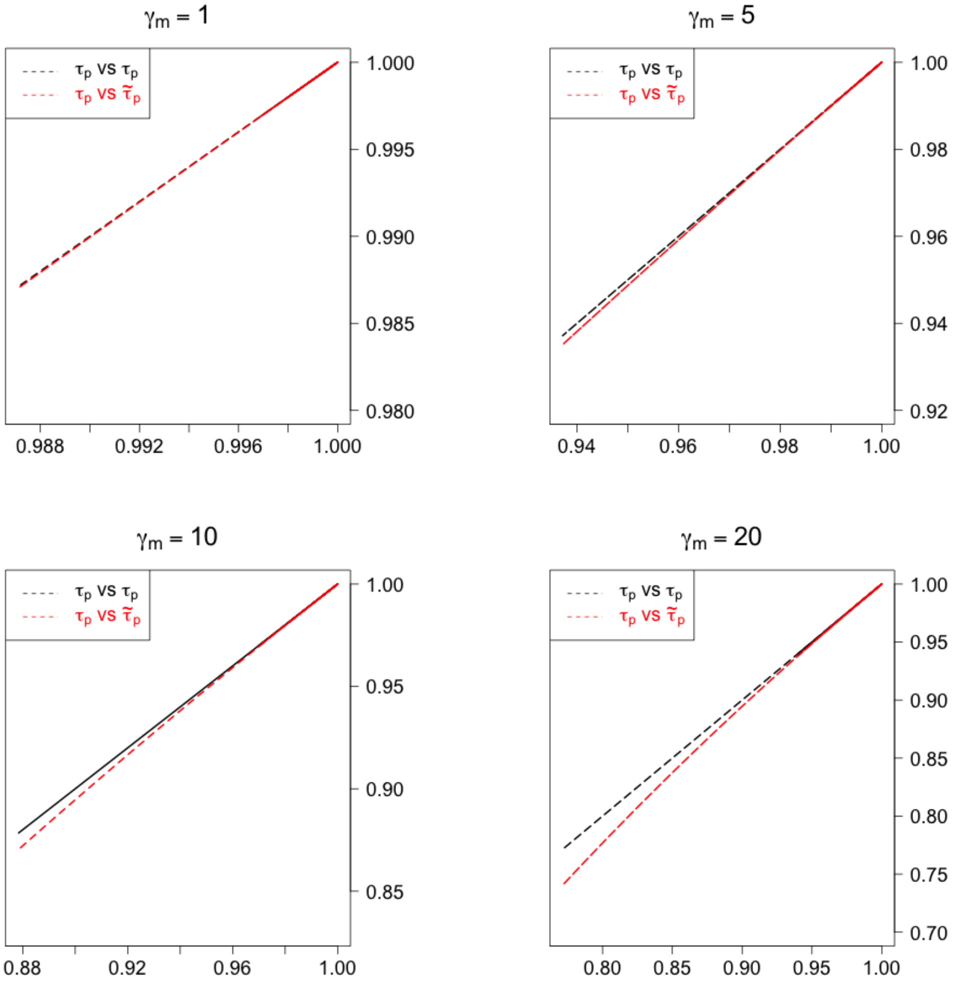


Figure 1.5: Broken black line plots $\tau_p(\lambda)$ against $\tau_p(\lambda)$ and broken red lines plots $\tau_p(\lambda)$ against its linear approximation, $\tilde{\tau}(\lambda)$. In subsequent sections we show that identification methods that assume linearity of Beer's law perform worse when $\gamma_m = 20$ versus when $\gamma_m = 5$, holding all other embedding parameters constant. This suggests that models that take into account non-linearity may improve results.

Also note that the concentration of chemical m can vary over the region of the embedded plume. For example, we could choose γ_m to be uniform over the embedded plume region. Or we can choose a Gaussian embedding shape where the concentration of chemical m is highest in the center of the plume and dissipates as the distance from the plume center increases. In the next section under subsection 1.5.1 we describe semi-synthetic datasets where we embedded plumes, and in these datasets the plumes are embedded such that γ_m is uniform over the embedding region.

The value of the plume temperature also affects how difficult it will be to detect and identify. If the temperature difference between the plume and background is small, the background signal is flat, and the plume is optically thin, then the signal captured by the sensor is approximately linear, which makes the tasks of detection and identification less challenging. Therefore, specifying the plume temperature, T_p to be around 5°K, will typically result in an easier embedding scenario. However, notice that in equation (1.9) that the temperature of the background is not a parameter that needs to be estimated. Therefore in practice we set the plume temperature to be a certain number of degrees greater or less than the estimated atmosphere temperature, T_a.

1.5 Hyperspectral datasets

1.5.1 Semisynthetic datasets

In this section we describe three partially-synthesized hyperspectral images that we simulated ourselves. The first semi-synthetic dataset we evaluate in this work embeds plumes into the plume-free hyperspectral cube shown in Figure 1.1. Figure 1.6 shows the spectral mean (over wavelengths) of the hyperspectral cube, and where we have geographically embedded plumes. Note that the hyperspectral image has a total of $J = 89,600$ pixels and $I = 85$ wavelengths in LWIR. We embedded plumes in areas that have relatively flat background, for a total of 7 plumes, one for each possible combination of three different chemicals, triethyl phosphate (TEP, commonly used in flame retardants and pesticides), dimethyl methylphosphonate (DMMP, also used in flame retardants as well as in the manufacture of chemical weapons), and sarin (which is classified as a weapon of mass destruction according to UN Resolution 687). Figure 1.7 plots each chemical's spectral signature, and Table 1.1 shows the correlation and cosine similarity among the three chemicals. TEP and Sarin are relatively uncorrelated, and therefore, we would expect a low occurrence of identification confusion between these two chemicals. However, we would expect a higher rate of identification confusion between TEP and DMMP, since their spectral signatures are similar. Finally, in the embedding algorithm we set the temperature of the plume to be 5°K hotter than the atmosphere's estimated temperature, and we set $\gamma_m = 5$ uniformly over the plume region if chemical m was present in the plume (and $\gamma_m = 0$ otherwise). These values ensure that the signals from each pixel are approximately linear. Therefore in the sequel we refer to this dataset as the closely linear dataset.

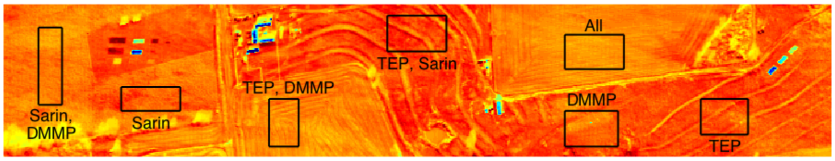


Figure 1.6: Partially synthesized hypersepctral image, with simulated plume regions marked, and chemicals embedded noted for closely linear dataset

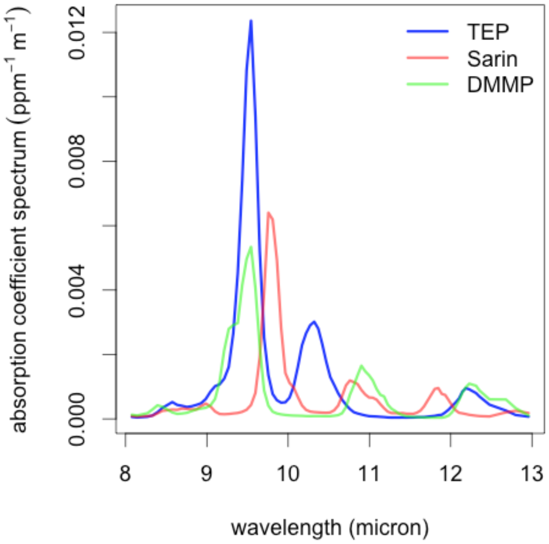


Figure 1.7: Interpolated spectral signatures of TEP, Sarin, and DMMP.

Correlation coefficient

	TEP	Sarin	DMMP
TEP	1.0000	0.0077	0.8220
Sarin		1.0000	-0.0553
DMMP			1.0000

Cosine similarity

	TEP	Sarin	DMMP
TEP	1.0000	0.2501	0.8663
Sarin		1.0000	0.2421
DMMP			1.0000

Table 1.1: Linear correlation and cosine similarity among the three chemicals used in our synthesized dataset.

The second semi-synthetic dataset that we will evaluate is unlike the first because we will consider chemicals that are not actually present in the hyperspectral image. In this respect this dataset is more like the challenge dataset that we describe more in the next section, Section 1.5.2. The challenge dataset is meant to act as our "real" dataset even though the plumes are also embedded in this image, but we did not do the embedding ourselves (recall that real plume release hyperspectral images are expensive to capture and difficult to obtain due to necessary security clearances). Therefore since this second semi-sythetic dataset is more like the challenge dataset, we will refer to it as the "realistic" dataset. Figure 1.8 shows the embedding truth for the realistic dataset. We embed one plume with TEP and DMMP which are highly correlated, and a second plume with TEP and Sarin, which are less correlated (see Table 1.1). However, unlike the closely linear dataset, which considered only TEP, Sarin, and DMMP as possible chemicals, for the realistic dataset we consider TEP, Sarin, DMMP as well as two additional chemicals, 1,1-Difluoroethane (DFE), and ammonia. DFE is used in refrigerants, aerosol sprays, and automobiles. Ammonia can be found in fertilizer, cleaners, fuel, and lifting gases used to fill weather balloons. We plot the spectral signatures for DFE and ammonia in Figure 1.9, and show the correlation coefficients among all five chemicals in Table 1.2. Lastly, like the closely linear dataset, we embed the two plumes such that linear assumptions hold.

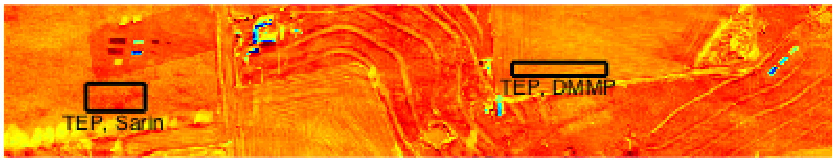


Figure 1.8: Partially synthesized hypersepctral image, with simulated plume regions marked, and chemicals embedded noted for realistic dataset

	TEP	**Sarin**	**DMMP**	**DFE**	**Ammonia**
TEP	1.0000	0.0077	0.8220	-0.1019	0.1595
Sarin		1.0000	-0.0553	-0.0975	0.0425
DMMP			1.0000	-0.1491	0.0742
DFE				1.0000	0.0223
Ammonia					1.0000

Table 1.2: Linear correlation among 5 chemicals considered in the realistic dataset

The third semi-synthetic dataset that we will evaluate will help us explore how our models perform when the signal captured by the sensor is not linear, and specifically when the embedded plumes are not optically thin. This non-linear dataset is similar to the closely linear dataset. We embed plumes on the same plume free image with the same number of plumes in the same locations, and with the same combinations of chemicals as shown in Figure 1.6. We also embed the plumes to be 5°K hotter than the estimated atmosphere temperature. However, we specify $\gamma_m = 20$ uniformly over the plume region, if chemical m was present in the plume, and $\gamma_m = 0$ otherwise. In the sequel we refer to this dataset as the non-linear dataset.

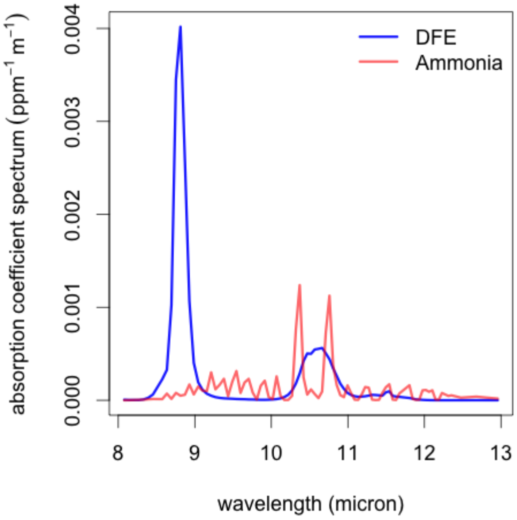


Figure 1.9: Interpolated spectral signatures of DFE and Ammonia.

1.5.2 MIT Lincoln Laboratory Challenge Dataset

In April 2014, MIT Lincoln Laboratory released a set of six hyperspectral images with different chemical plumes embedded in them. These images were provided as an open challenge to the research community interested in developing identification algorithms. The datasets are

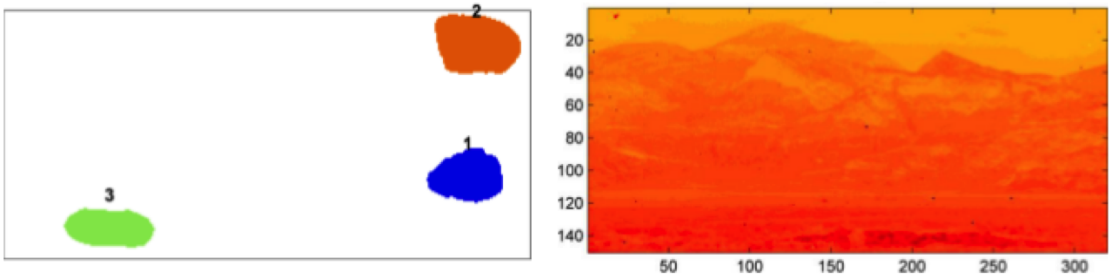


Figure 1.10: Embedding truth corresponding to challenge data cube (left) and spectral mean of MIT Lincoln Lab challenge data cube.

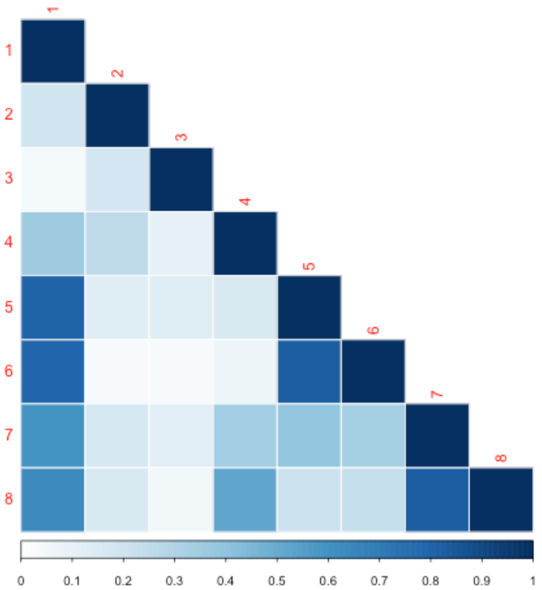


Figure 1.11: Absolute value of the pairwise correlations for the spectral signatures of MIT Lincoln Lab challenge data.

available to the wider research community at `https://grassmann.math.colostate.edu/ATD/ChemIDChallengeTrain.html`. We consider these datasets "real" data since we did not simulate them ourselves. In this work we focus on one of the more difficult cubes in the challenge, "GID_Targ_Mix_JH", whose spectral mean is shown in the right panel of Figure 1.10. The dimensions of this cube are $150 \times 320 \times 129$ for a total of $J = 48,000$ pixels, with measurements over $I = 129$ wavelengths in LWIR. Note that this cube is side-looking, as opposed to the downward-looking geometry of the partially synthesized datasets described in the previous section. For this challenge dataset there are 8 possible chemicals numbered 1-8. In the left panel of Figure 1.10 we show the embedding truth of plume locations, where plume 1 contains chemicals 1 and 3, plume 2 contains chemicals 5 and 6, and plume 3 contains chemicals 7 and 8. The absolute value of the pairwise correlations among these chemicals is shown in Figure 1.11. We expect that identifying plumes 2 and 3 in the challenge data cube will be especially difficult, since chemicals 5 and 6 are highly correlated and chemicals 7 and 8 are also highly correlated. However, we expect that identification of plume 1 will be relatively easy as chemical 3 has little correlation with chemical 1. Note that in the sequel we refer to this dataset as the challenge dataset.

1.6 Performance metrics

To evaluate the performance of our algorithms we use multi-class classification performance metrics. Therefore, we present detection and identification results using confusion matrices, which are square matrices that tally the number of pixels correctly and incorrectly classified over all chemical classes. Confusion matrices with more non-zero off-diagonal elements are considered poorer classifiers. Metrics that can summarize a confusion matrix into one number are useful, especially when the number of possible classes is large. In particular, we focus on the accuracy (ACC), and Matthews correlation coefficient (MCC). ACC takes the ratio of the number of correctly classified pixels over the total number of pixels, and can take on values in the range [0,1], where a value close to 1 implies a good identifier, and a value close to 0 implies a poor identifier. The MCC is a multi-class extension of Pearson's correlation coefficient (Gorodkin, 2004) that can take on values in the range [-1,1]. Smilar to ACC, values of MCC close to 1 imply a good identifier.

1.7 Parellel computation and MCMC convergence monitoring

All of our identification algorithms can be implemented on the hyperspectral datasets in parallel over pixels to some degree depending on the identification method. For all of the identification model implementations we use openMP in C++ on all cores of a 64-core Linux server. Furthermore, for Bayesian methods which require Markov Chain Monte Carlo (MCMC), we monitor convergence of the MCMC by assessing the (unnormalized) log posterior distribution.

Chapter 2

Supervised linear variable and model selection methods for remotely sensing plumes in hyperspectral imagery

In this chapter we review methods that approximate the relationship between the spectrum associated with a pixel and the spectrum of chemicals by a linear regression, and treat the detection/identification problem as one of variable or model selection. First in Section 2.1 we discuss linear assumptions that justify the use of a linear likelihood, which we derive in Section 2.2. Then in Sections 2.3, and 2.4 we outline the least absolute shrinkage and selection operator (LASSO), and Bayesian indicator model selection (BIMS) as techniques for detection and identification. Then in Section 2.5 we present the detection and identification results of each method on the closely linear dataset. Lastly, in Section 2.6 we discuss and compare the overall performance of both methods, and their shortcomings.

2.1 Linear assumptions

Under some physically plausible assumptions, we can simplify the non-linear phsyical model from Section 1.2. As mentioned previously, if the plume is optically thin (i.e., $\tau_p(\lambda)$ is close to 1) then we can assume that $\sum_{m=1}^{M} \gamma_m \alpha_m(\lambda) \ll 1$ for all λ, and we can linearize Beer's law by using the first term of its Taylor expansion,

$$\tau_p(\lambda) \approx 1 - \sum_{m=1}^{M} \gamma_m \alpha_m(\lambda).$$

Furthermore, if we assume the background signal is flat (i.e., that it does not have sharp features), then $L_b(\lambda) \approx B(\lambda, T_b)$. Lastly, if the absolute difference in temperature between the plume (T_p) and background (T_b) is small (usually less than 5°K), then we can assume $B(\lambda, T_p) - B(\lambda, T_b) \approx C_B \Delta(T_p - T_b)$, where C_B is a constant independent of wavelength and temperature. Under these assumptions and after some algebra, the on-plume radiance signal can be approximated as,

$$L_{\text{on}}(\lambda) \approx \sum_{m=1}^{M} \{C_B \Delta T \gamma_m\}\{\tau_a(\lambda)\alpha_m(\lambda)\} + L_{\text{off}}(\lambda). \tag{2.1}$$

This equation can be understood as a "signal-plus-clutter" model, where the background signal, $L_{\text{off}}(\lambda)$, is the clutter.

Hyperspectral sensors capture the at-sensor radiance signal as shown in equation (2.1) on a series of I adjacent spectral bands centered at bandwidths $\lambda_1, \ldots, \lambda_I$ and then processes it through its spectral response function, $R_F(\lambda)$. The sensors then introduce noise, $n(\lambda)$, to produce the pixel spectrums in a hyperspectral image $\boldsymbol{x} = (x_1, \ldots, x_n)$. It follows that

$$x_i = \sum_{m=1}^{M} g_m s_{m,i} + v_i, i = 1, \ldots, I, \tag{2.2}$$

where the unknown regression coefficients, $g_m = C_B \Delta T \gamma_m$, are proportional to γ_m, which is the concentration of the chemical. The covariates $s_{i,m} = \{[\tau_a(\lambda_i)\alpha_m(\lambda_i)] * R_{\text{F}}(\lambda_i)\}$ correspond to the spectral chemical signatures, the error term $v_i = \{L_{\text{off}}(\lambda_i) * R_{\text{F}}(\lambda_i)\} + n(\lambda_i)\}$ is the background signal plus sensor noise, and $*$ denotes the convolution operator.

The linear model in (2.2) provides a general framework in which the detection, identification, and quantification tasks can be cast. In particular, note that detection involves testing whether

$\boldsymbol{g} = (g_m, \ldots, g_m)$ is different from the zero-vector, identification involves testing which individual elements in $\boldsymbol{g}$ are non-zero, and quantification involves estimating the exact values of the elements in $\boldsymbol{g}$.

2.2 Linear Likelihood

Let $x_{j,i}$ be the measurement from the hyperspectral sensor in pixel $j = 1, \ldots, J$ at bandwith λ_i, $i = 1, \ldots, I$ and $s_{i,m}$ be the hyperspectral signature of chemical $m = 1, \ldots, M$ at the same bandwith i. The structure of (2.2) suggests a likelihood of the form,

$$\boldsymbol{x}_j = \boldsymbol{S}\boldsymbol{g}_j + \boldsymbol{v}_j, \qquad \boldsymbol{v}_j \sim \mathsf{N}\left(\boldsymbol{\mu}, \sigma_j^2 \boldsymbol{\Sigma}\right), \qquad j = 1, \ldots, J,$$

where we assume the background signal (plus sensor noise) follows a normal distribution. The parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ correspond to the mean and variance structures associated with the background. In this manuscript we focus on supervised problems in which we need to detect/identify chemicals from a finite set of pre-specified, application-specific candidates. Hence, the model above is often referred to as a linear un-mixing model, because the goal is to un-mix the signal into a set of chemicals.

For computational efficiency we proceed by replacing $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ with point estimates $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Sigma}}$. In particular, $\hat{\boldsymbol{\mu}}$ corresponds to the mean signal over all pixels, while $\hat{\boldsymbol{\Sigma}}$ is robustly calculated from the whole image using a dominant mode rejection procedure (Manolakis et al., 2009), which uses only dominant eigenvalues and diagonal loading to ensure invertibility of $\hat{\boldsymbol{\Sigma}}$. As a consequence, we can rewrite the observational model as

$$\boldsymbol{y}_j = \boldsymbol{Z}\boldsymbol{g}_j + \boldsymbol{\epsilon}_j, \qquad \boldsymbol{\epsilon}_j \sim N(\boldsymbol{0}, \sigma_j^2 I), \qquad j = 1, \ldots, J, \tag{2.3}$$

where $\boldsymbol{y}_j = \hat{\boldsymbol{\Sigma}}^{-1/2}(\boldsymbol{x}_j - \hat{\boldsymbol{\mu}})$, $\boldsymbol{Z} = \hat{\boldsymbol{\Sigma}}^{1/2}\boldsymbol{S}$, and $\boldsymbol{\epsilon}_j = \hat{\boldsymbol{\Sigma}}^{-1/2}\boldsymbol{v}_j$.

2.3 LASSO as state-of-the-art variable selection method for detection and identification

Frequentist approaches to hypothesis testing based on p-values (Fisher, 1925) are widespread. However, such methods require addressing the issue of multiple comparisons. Another variable selection approach from the frequentist literature is the LASSO (Tibshirani, 1996), which belongs to the class of penalized least squares methods. Other models that belong to this class are the elastic net (Zou and Hastie, 2005), and the smoothed clipped absolute deviation (Fan and Li, 2001). The LASSO is a popular variable selection method because it is easy and fast to compute. The LASSO performs variable selection by adding an ℓ_1 penalty on the coefficients of a linear model, such as the one described in Section 2.2. Becasue an ℓ_1 penalty is non-differentiable at zero, we can obtain exact zero estimates for the regression coefficients. In terms of detection and identification, if a coefficient is estimated as zero, then the corresponding chemical is not present in the plume. Furthermore if all coefficients are estimated as zero, then the signal is composed of only the atmosphere and background. The level of shrinkage to zero in the LASSO coefficient estimates is controlled by a scalar value, which is referred to as the LASSO penalty parameter. A LASSO penalty parameter equal to zero will lead to the ordinary least squares estimate, which can never be zero. Larger LASSO penalty parameter values will force more coefficients to be excluded from the model.

From an estimation point of view, the LASSO has a Bayesian interpretation. Specifically, the LASSO's ℓ_1 penalty corresponds to an unnormalized double exponential prior, and the corresponding point estimates are the the posterior mode of the model. However, as model selection methods, the LASSO cannot be considered a Bayesian procedure.

2.3.1 The LASSO method

For each pixel, independent estimates of the vectors of regression coefficients are obtained by solving,

$$\min_{\boldsymbol{g}_j} ||\boldsymbol{y}_j - Z\boldsymbol{g}_j||^2 + \eta_j ||\boldsymbol{g}_j||_1, \tag{2.4}$$

for each $j = 1, ..., J$, where each pixel has its own LASSO penalty parameter, η_j. An alternate approach would be to consider one common penalty parameter over all pixels, i.e., $\eta_1 = \cdots = \eta_J$. However, we have found that individual penalties for each pixel yields higher detection and identification accuracy.

As mentioned previously, η_j controls the amount of shrinkage to zero in the coefficient estimates in $\boldsymbol{g}_j$. There are several methods available for choosing the optimal value for η_j. A widely use method is K-fold cross validation (Tibshirani, 1996). Criterions, such as the Akaike Information Criterion (Akaike, 1973) can also be used. In our implementation we use the Bayesian Information Criterion, or BIC (Schwartz, 1978) to choose the optimal value for the penalty parameter. Specifically, we consider a grid of penalty parameter values, $\{\tilde{\eta}_j^{(k)}; k = 1, ..., K\}$, and choose the value that minimizes,

$$\frac{1}{\hat{\sigma}_j^2} ||\boldsymbol{y}_j - Z\tilde{\boldsymbol{g}}_j^{(k)}|| + log(I) M_j \left(\tilde{\eta}_j^{(k)} \right), \tag{2.5}$$

where $\hat{\sigma}_j^2 = I^{-1} \sum_{i=1}^{I} \left(y_{j,i} - \boldsymbol{z}_i' \hat{\boldsymbol{g}}_j \right)^2$, and $\hat{\boldsymbol{g}}_j = (\boldsymbol{Z}'\boldsymbol{Z})^{-1} \boldsymbol{Z} \boldsymbol{y}_j$ are the maximum likelihood estimates (MLE) of σ_j^2 and $\boldsymbol{g}_j$. The parameter, $\tilde{\boldsymbol{g}}_j^{(k)}$ is the LASSO estimate of $\boldsymbol{g}_j$ using penalty parameter, $\tilde{\eta}_j^{(k)}$. The grid of penalty parameters we use starts at $\tilde{\eta}_j^{(1)} = 0$, and the final value in the grid, $\tilde{\eta}_j^{(K)}$, will shrink all elements in $\boldsymbol{g}_j$ to be estimated as 0.

We use the split Bregman algorithm outlined in Ye and Xie (2011) to solve the LASSO problem in (2.4). The algorithm constrains the optimization problem and iteratively reaches the solution with the use of the augmented Lagrangian and soft thresholding. We explain the computational algorithm in more detail in Chapter 5.

2.4 Bayesian model selection approach to detection and identification

The fundamental tools of Bayesian model selection are Bayes factors and posterior model probabilities. Berger and Pericchi (2001) discuss several advantages of taking a Bayesian approach to model selection. For example, p-values are not consistent, i.e., their accuracy in evaluating evidence to support the null hypothesis does not increase with more data. In contrast, Bayes factors and posterior model probabilities are consistent. Also, unlike the classical approach, which treats model selection differently for two models (hypothesis testing) versus more than two models (variable selection), the Bayesian approach is essentially the same for any number of possible models. Scott and Berger (2006) show that Bayesian model selection does not have the multiple comparison problem that affects multiple hypothesis testing. Lastly, hypothesis testing and other non-Bayesian methods, like the LASSO, do not not take into account colinearity among predictors. Bayesian model selection can address colinearity through the use of certain priors.

One challenge of Bayesian model selection is that Bayes factors and posterior model probabilities require calculation of the marginal likelihood. Closed form solutions of the marginal likelihood are not always available. However, one way to deal with this is through reversible jump Markov chain Monte Carlo (Green, 1995). The BIC can also be used to provide an asymptotic approximation to the log of the marginal likelihood.

An alternative way to formulate Bayesian model selection is a method named by O'Hara and Sillanpaa (2009), "Bayesian Indicator Model Selection" (BIMS). In the BIMS method indicator variables for each regression coefficient are introduced. If a regression coefficient is different from zero, then the corresponding indicator variable takes on a value of 1, and 0 otherwise. Under this approach we are interested in making inference on the indicator variables. For example, if an indicator variable is equal to 1, then the corresponding chemical is present in the pixel, and is not present otherwise. Following standard Bayesian procedure, we place priors on the regression coefficients and on the indicator variables. Kuo and Mallick (1998) suggest placing independent priors on the regression coefficients and on the indicator variables, whereas Carlin and Chib (1995) propose making the two sets of hyperparameters dependent in the prior. The method of stochastic search variable selection (George and McCulloch, 1993) also suggests dependence through the use of spike and slab priors.

In our implementation of BIMS we place a mixture of g-priors (Liang et al., 2008; Zellner, 1986) on the regression coefficients given the indicator variables. This prior takes into account colinearity among predictors. Also for linear models this prior is computationally advantageous, because we can avoid posterior sampling of the regression coefficients. We perform computations for this model using an MCMC algorithm that updates blocks of parameters sequentially. This algorithm is implemented in a parallel fashion to speed up computation. For details of the algorithm see Chapter 5.

2.4.1 The BIMS method

To describe the model we introduce, for each pixel $j = 1, \dots, J$, a set of latent indicator variables $\boldsymbol{\omega}_j = (\omega_{1,j}, \dots, \omega_{M,j})'$ such that $\omega_{m,j} = 0$ if $g_{m,j} = 0$ and $\omega_{m,j} = 1$ otherwise. Because our focus is on detection and identification problems, the vectors $\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_J$ are the main parameters of interests, and the actual values of the regression coefficients $\boldsymbol{g}_1, \dots \boldsymbol{g}_J$ can be considered nuisance parameters.

Conditionally on the indicator vector $\boldsymbol{\omega}_j$ and the variance σ_j, we assign the vector of regression coefficients, $\boldsymbol{g}_j$, a mixture of g-priors,

$$\boldsymbol{g}_j | c_j, \sigma_j^2, \boldsymbol{\omega}_j \sim \mathsf{N}\left(\boldsymbol{g}_j(\boldsymbol{\omega}_j) \mid \boldsymbol{0}, c_j \sigma_j^2 \left\{\boldsymbol{Z}'(\boldsymbol{\omega}_j)\boldsymbol{Z}(\boldsymbol{\omega}_j)\right\}^{-1}\right) \prod_{m:\omega_{m,j}=0} \delta_0(g_{m,j}), \tag{2.6}$$

where $\boldsymbol{g}_j(\boldsymbol{\omega}_j)$ denotes the subvector of $\boldsymbol{g}_j$ that corresponds to the entries where $\omega_{m,j} = 1$, $\boldsymbol{Z}(\boldsymbol{\omega}_j)$ is the submatrix of $\boldsymbol{Z}$ that only includes the columns where $\omega_{m,j} = 1$, $\delta_0(\cdot)$ denotes the degenerate measure at zero, and $c_j \sim \mathsf{IGam}(.5, .5I)$. On the other hand, the variances $\sigma_1^2, \dots, \sigma_J^2$ are assigned independent Jeffreys priors, $p(\sigma_j^2) \propto \frac{1}{\sigma_j^2}$.

Note that the proposed priors for the variances $\sigma_1, \dots, \sigma_J$ are improper. However, because these parameters are shared across all potential models it is well known that they result in well-calibrated posterior probabilities (e.g., see Berger and Pericchi, 2001). Furthermore, their use along with that of mixtures of g-priors for $\boldsymbol{g}_j \mid \sigma_j^2$ has both computational and theoretical advantages. From a computational perspective, these choices allow us to explicitly integrate out the variances, σ_j^2, and the regression coefficients, $\boldsymbol{g}_j$,

$$p(\boldsymbol{y}_1, \dots, \boldsymbol{y}_J \mid \boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_J, c_1, \dots, c_J) \propto$$
$$\prod_{j=1}^{J} (c_j + 1)^{-\sum_{m=1}^{M} \omega_{m,j}/2} \left(\boldsymbol{y}_j' \left[I - \frac{c_j}{c_j + 1} \boldsymbol{Z}'(\boldsymbol{\omega}_j) \left\{\boldsymbol{Z}'(\boldsymbol{\omega}_j)\boldsymbol{Z}(\boldsymbol{\omega}_j)\right\}^{-1} \boldsymbol{Z}(\boldsymbol{\omega}_j) \right] \boldsymbol{y}_j \right)^{-I/2}, \tag{2.7}$$

which helps us avoid the need for Reversible Jump MCMC algorithms. From a theoretical perspective, letting c_j be random in the prior addresses the information paradox (Bartlett, 1957) and Barlett's paradox (Berger and Pericchi, 2001). Furthermore, unlike LASSO penalties, the g-prior incorporates information on the correlations among predictors, which should help ameliorate any complication arising from multicolinearity.

The model is completed by specifying a prior on the indicator vectors $\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_J$. Following Scott et al. (2010) we focus on priors of the form

$$p(\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_J) = \prod_{j=1}^{J} \frac{\Gamma\left(a + \sum_{m=1}^{M} \omega_{m,j}\right) \Gamma\left(b + J - \sum_{m=1}^{M} \omega_{m,j}\right)}{\Gamma\left(a+b+J\right)}, \tag{2.8}$$

i.e., we assume that the number of chemicals present in pixel j follows a Beta-Binomial distribution with mean $a/(a+b)$. The structure of this prior can be derived by assuming that each chemical is present on a given pixel independently with unknown probability θ_j, which is in turn assigned a beta prior with parameters a and b. In our evaluations we set $a = b = 1$, which is equivalent to assuming uniform priors on the θ_j's. The fact that θ_j is assigned a prior rather than fixed in advance induces correlation between $\omega_{m,j}$ and $\omega_{m',j}$, which in turn allows the model to automatically adjust for multiplicities. Using posterior samples of $\boldsymbol{\omega}_j$, we then identify pixel j corresponding to the chemical model that yields the maximum posterior probability among all chemical models. More precisely, recall that $\boldsymbol{\omega}_j = (\omega_{1,j}, ..., \omega_{M,j})$, and $\omega_{m,j} = 1$ if chemical m is present in pixel j, and $\omega_{m,j} = 0$, otherwise. Let $\boldsymbol{\omega}_j^{(p)}$ denote the p-th possible chemical model for $\boldsymbol{\omega}_j$, where there are $p = 1, ..., 2^M$ possible chemical models.

$$F_j^{(p)} = \sum_{k=1}^{K} 1_k(\boldsymbol{\omega}_j = \boldsymbol{\omega}_j^{(p)})$$

is the frequency at which model $\boldsymbol{\omega}_j^{(p)}$ was visited for $\boldsymbol{\omega}_j$ over $k = 1, ..., K$ posterior MCMC samples (after convergence) where K is large. The chemical model, $\boldsymbol{\omega}_j^{(p)}$, that yields the highest frequency, i.e., $\max_p F_j^{(p)}$, will be the chemical model with the highest posterior probability for $\boldsymbol{\omega}_j$, and will be the identified chemical model for pixel j. This strategy can be used for any value of M, i.e., for any number of possible chemical models.

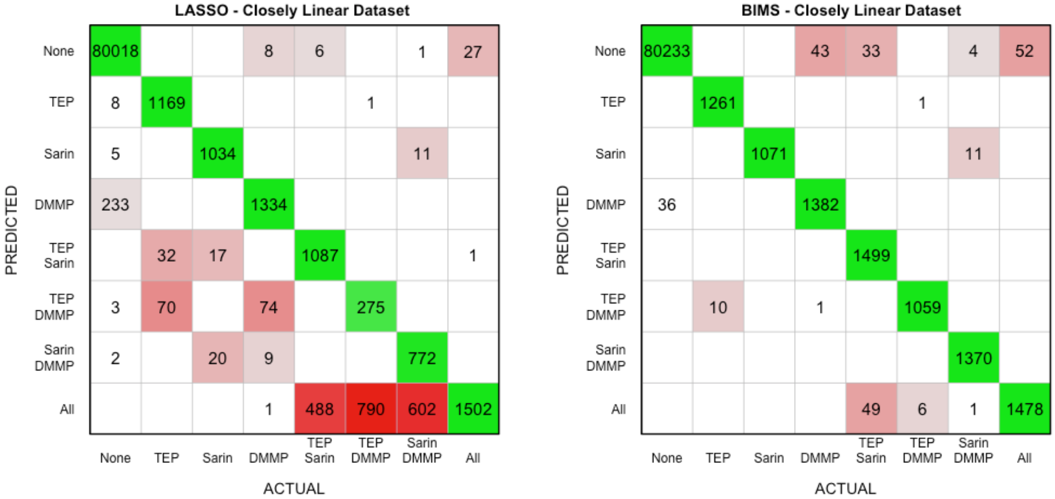
In this work we follow the school of thought as shown in Scott et al. (2010) that taking a Bayesian approach of placing priors on inclusion probabilities of predictors and on other model parameters (i.e., the mean, variance, coefficients, etc.) will alleviate the issues of multiple comparisons in model selection problems. However, we do recognize another school of thought where false discovery rate (FDR) rules are needed to properly account for multiple comparisons in Bayesian model selection methods (Muller et al., 2006; Wakefield, 2007; Whittemore, 2007). FDR rules are especially important for Bayesian approaches where massive amounts of multiple comparisons arise especially in gene expression data. However, FDR implicitly assumes that something is significant, for example, in neuroscience FDR would assume that points in the same region of the brain are activated during an experiment. However, this type of logic does not fit in well with the application at hand, since we cannot assume a plume is definitely present somewhere in the image.

For our choice of prior on c_j we followed Liang et al. (2008) closely where they performed prior sensitivity analysis by exploring several alternative priors for c_j. In particular they explored Empirical Bayes (EB), and the hyper g-prior. They also evaluated the Zellner-Siow (ZS) prior, which is the prior we use in our implementation. They prove that the ZS prior is consistent under the null model, whereas EB and the hyper g-prior are not. They also show that the ZS prior was robust to misspecification of c_j and permits fast marginal likelihood calculations. As for the prior we chose for ω_j, it is a well established prior on models that avoids favoring medium sized models. Furthermore we recommend $a = 1$ and $b = 1$ in the Beta-Binomial prior as default values, unless substantive prior information about the likelihood of a chemical is available. We did perform prior sensitivity analysis by exploring alternative hierarchical specifications for this prior (e.g., letting the prior inclusion probability be the same for all pixels and gases, as well as letting it vary with the gas rather than the pixel identity), but (2.8) yielded the best empirical results.

2.5 LASSO and BIMS results on closely linear semi-synthetic dataset

In this section we evaluate the performance of the LASSO and BIMS method on the semi-synthetic image that is closely linear (see Section 1.5). When fitting the models we assume that TEP, Sarin, DMMP are the only possible chemicals that could be in the image. Figure 2.1 displays the confusion matrices for each of the methods, and Table 2.1 shows the ACC and MCC values for each of the methods.

In general BIMS performs better than the LASSO as it has higher ACC and MCC values. Looking at the confusion matrices, the LASSO does relatively well in classifying pixels with “None”, “All”, and one chemical classes. However, it tends to overestimate the number of chemicals present in plumes that have 2 chemicals. BIMS on the other hand is able to classify



LASSO - Closely Linear Dataset

PREDICTED \ ACTUAL	None	TEP	Sarin	DMMP	TEP Sarin	TEP DMMP	Sarin DMMP	All
None	80018			8	6		1	27
TEP	8	1169				1		
Sarin	5		1034				11	
DMMP	233			1334				
TEP Sarin		32	17		1087			1
TEP DMMP	3	70		74		275		
Sarin DMMP	2		20	9			772	
All				1	488	790	602	1502

BIMS - Closely Linear Dataset

PREDICTED \ ACTUAL	None	TEP	Sarin	DMMP	TEP Sarin	TEP DMMP	Sarin DMMP	All
None	80233			43	33		4	52
TEP		1261				1		
Sarin			1071				11	
DMMP	36			1382				
TEP Sarin					1499			
TEP DMMP		10		1		1059		
Sarin DMMP							1370	
All					49	6	1	1478

Figure 2.1: Confusion matrices for the LASSO and BIMS.

	LASSO	BIMS
ACC	0.9311	0.9982
MCC	0.8654	0.9859

Table 2.1: ACC and MCC values for the simulated dataset.

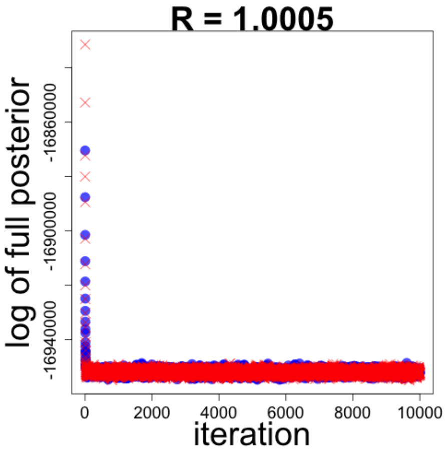


Figure 2.2: Trace plot of log posterior distribution under BIMS for the closely linear dataset for 2 MCMC chains, where one chain is represented by red x's, and the second chain is represented by blue dots. Using the first 5,000 iterations it each chain, a Gelman and Rubin statistic of 1.0005 was calculated, implying convergence was reached.

2-chemical pixels with higher accuracy, even for the TEP and DMMP plume, where these two chemicals have highly correlated spectral signatures. In fact BIMS classifies more pixels correctly than the LASSO over all classes, except the "All" class. Indeed, while the LASSO makes substantially more false positive mistakes than BIMS (i.e., it tends to identify more chemicals than are really present), BIMS makes slightly more false negative mistakes than the LASSO (in the sense of identifying pixels that are chemical-present as background pixels). Therefore the LASSO does a decent job of detection, i.e., detecting where plume is and is not, but struggles to identify the correct chemical structure in plumes.

Since both the LASSO and BIMS assume independence among pixels, we can implement these methods over all pixels in parallel allowing for fast processing times. For the closely linear dataset, the LASSO took 3.54 minutes to solve. Recall that we solve the LASSO over a grid of possible LASSO penalty values and use the BIC to choose the optimal LASSO value for each pixel (see Section 2.3.1). In particular we considered 59 LASSO penalties from 0 to 5.8, where 5.8 shrank all coefficients to 0. Therefore in 3.54 minutes the LASSO was solved 59 times for a total of 89,600 pixels. As the number of pixels increases, solve time for the LASSO will also increase, but not dramatically due to the use of parallel computing.

The BIMS method requires implementation of an MCMC algorithm to obtain posterior samples. We ran the MCMC for 10,000 iterations with a burn-in of 5,000 iterations for inference. It took 2.8 hours or 1.03 seconds per iteration. Considering that there are 89,600 pixels in the image, parallel computing greatly increases the speed of the BIMS algorithm. We note that there is room for some improvements in the C++ code to increase efficiency and speed further. We ran a second MCMC chain for 10,000 iterations, and we show the log posterior distribution values over iterations for both chains in Figure 2.2. Using the first 5,000 iterations in each of the two MCMC chains we obtained a Gelman and Rubin R-statstic of 1.0005, where a value of 1.2 or less implies convergence. The time needed to complete one iteration of BIMS over all pixels will increase as the number of pixels in an image increases, but like the LASSO, not dramatically due to the use of parallel computing, and this will be true for any hyperspectral image. However, increasing the number of chemicals that we consider as possibly present in the image will have a bigger effect on runtime. This is true for BIMS, LASSO, as well as for other spatial and nonparametric methods that we discuss in future sections. Since BIMS is a fairly simple Bayesian model, convergence of the MCMC algorithm is fast. In future sections we will describe more complex Bayesian models used for identification that require more time to reach convergence.

2.6 Discussion

In this chapter we evaluated supervised model selection methods that assumed the pixel signals in an image are linear. Futhermore, we implemented the methods on a dataset that was partially synthesized to have pixel signals that were close to linear. Despite this, the LASSO method struggled to identify chemicals correctly, however it was able to detect gas pixels from no-gas pixels relatively well. Although the LASSO is used as a variable selection method, at it's core it is truly an estimation method. In addition, the LASSO treats coefficients as independent and it does not take into account co-linearity among spectral signatures. To remedy these issues we turned to a Bayesian approach, BIMS, that takes into account model uncertainty, and through the use of g-priors accounted for colinearity among chemicals. This explains the higher classification accuracy obtained by BIMS. However, both of these methods share a common weakness as illustrated by Figure 2.3, which maps the detection and identification results of the closely linear semi-synthetic dataset on the area covered by the original image. The classification mistakes committed by both the LASSO and BIMS appear more or less random in the image. Indeed, both methods are implemented pixel-by-pixel, i.e., independence among pixels is assumed. This suggests that models that take into account spatial structure in the image,

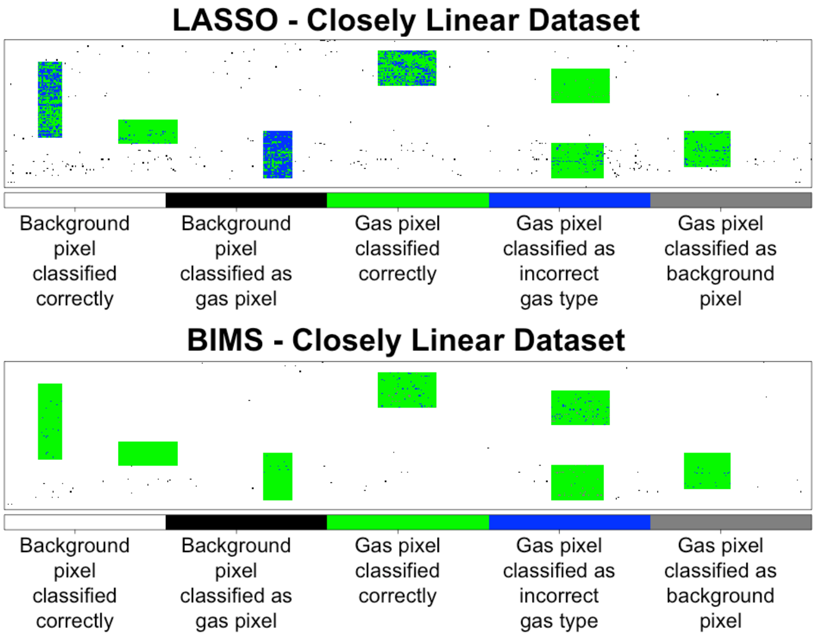


Figure 2.3: Identification results mapped onto the area covered by the synthesized data.

and especially in plumes, can improve detection and identification results.

Chapter 3

Incorporating spatial information in hyperspectral images

In this chapter we develop methods that are essentially spatial extensions of the LASSO and BIMS approaches discussed in Chapter 2. For these spatial models pixels are not independent, and instead it is assumed that neighboring pixels have similar signals. We justify this approach by the fact that gas plumes are spatially continuous, i.e., pixels that have plume tend to be adjacent. In Section 3.1 we explain the Fused LASSO, which is a spatial extension of the LASSO, as a detection and identification algorithm. Then in Section 3.2 we develop a novel Bayesian model selection method that incorporates spatial information through the use of Markov Random Fields (MRF). We then evaluate the performance of these methods in Section 3.4 on the closely linear dataset. We also evaluate each method on the challenge dataset and compare them to their pixel-by-pixel counterparts, the LASSO and BIMS, in Section 3.6. Lastly in Section 3.7 we discuss strengths and weaknesses of the methods.

3.1 Adding spatial information to the LASSO

Continuity of plumes has been previously exploited in the literature by incorporating fused LASSO penalties (Tibshirani et al., 2005b), also known as total variation norm penalties (e.g., see Guo et al., 2009; Iordache et al., 2012). Like the LASSO, the FL is a penalized regression method.

The FL generalizes the regular LASSO by placing an ℓ_1 penalty on regression coefficients, and a second ℓ_1 penalty on the differences between neighboring coefficients. Note that unlike the LASSO, the FL treats pixels as dependent through one common LASSO penalty over all observations. Furthermore, the FL also enforces spatial continuity through the FL penalty.

Since both the LASSO and FL penalties are non-differentiable at zero, we can obtain exact zero estimates for the coefficients as well as exact zero difference estimates between neighboring coefficients. In principle each penalty has different effects on the coefficients. The LASSO penalty controls how aggressively the algorithm will exclude chemicals from the model, where larger LASSO penalty values lead to all coefficients to be shrunk to zero. The FL penalty on the other hand controls how much information is borrowed between neighboring pixels. Large FL penalty values will lead to all coefficients to be estimated as equal to each other producing a smoothing effect. However, the two penalties interact with each other. For example, very large FL penalty values can cause equal and zero coefficient estimates even with moderate LASSO penalty values. Therefore, the FL performs variable selection while simultaneously forcing neighboring pixels to be similar.

Land and Friedman (1996) explored penalizing coefficient differences, but without a penalty on the regression coefficients themselves. The FL was popularized by Tibshirani et al. (2005a), and has been used in many applications, such as change-point detection in temporal data (Rojas and Wahlberg, 2014). In this set up a neighboring observation would be an observation in the time period before or after. For the case of imagery data, a neighborhood needs to be defined over pixels, which induces a two-dimensional space. Thus, for a first order neighborhood, a pixel's neighbors would be those directly to the left, to the right, above, and below. Higher order neighborhoods can be considered, however we focus only on the first order version. Neighborhood information is contained in the $J \times J$ neighborhood matrix, $\boldsymbol{W}$. More specifically, the estimators are defined by the solution to the following optimization problem,

$$\min_{\boldsymbol{g}_1,\cdots,\boldsymbol{g}_J} \sum_{j=1}^{J} \sum_{i=1}^{I} \left(y_{j,i} - \boldsymbol{z}_i'\boldsymbol{g}_j\right)^2 + \xi \sum_{j=1}^{J} \sum_{m=1}^{M} |g_{m,j}| + \zeta \sum_{j=1}^{J} \sum_{j'=1,j'\neq j}^{J} \sum_{m=1}^{M} w_{j,j'} \left|g_{m,j} - g_{m,j'}\right|, \tag{3.1}$$

where $w_{j,j'} = 1$ if pixel, j', is a first order neighbor to pixel, j, otherwise $w_{j,j'} = 0$. The parameter, ξ is the common LASSO penalty, and ζ is the FL penalty. When $\zeta = 0$, (3.1) is equivalent to the LASSO problem (see equation (2.4)) with $\eta_1 = ... = \eta_J = \xi$. Note that unlike the LASSO, the FL is a non-convex optimization problem. One novelty of our approach is that we solve the FL problem using the split Bregman algorithm (Goldstein and Osher, 2009; Osher et al., 2005). In particular we use the split Begman for large scale FL as outlined in Ye and Xie (2011). We discuss the algorithm in more detail in Chapter 5.

We perform a two-step process to choose ξ, and ζ. First, we ignore ζ, and find the optimal value of ξ over a grid of values, $\tilde{\xi}^{(1)}, ..., \tilde{\xi}^{(K)}$, starting at $\tilde{\xi}^{(1)} = 0$ to the final value, $\tilde{\xi}^{(K)}$, which forces all coefficients in all pixels to be 0. Similar to the implementation of the LASSO, we choose the value of ξ that minimizes the BIC,

$$\frac{1}{\hat{\sigma}^2} \sum_{j=1}^{J} \sum_{i=1}^{I} \left(y_{j,i} - \boldsymbol{z}_i' \tilde{\boldsymbol{g}}_j^{(k)} \right)^2 + log(I * J) M(\tilde{\xi}^{(k)}), \tag{3.2}$$

where $\hat{\sigma}^2 = (I * J)^{-1} \sum_{j=1}^{J} \sum_{i=1}^{I} \left(y_{j,i} - \boldsymbol{z}_i' \hat{\boldsymbol{g}}_j \right)^2$, and $\hat{\boldsymbol{g}}_j = (\boldsymbol{Z}'\boldsymbol{Z})^{-1} \boldsymbol{Z}' \boldsymbol{y}_j$, are the MLE's of σ^2 over all pixels, and of $\boldsymbol{g}_j$, respectively. The parameter, $\tilde{\boldsymbol{g}}_j^{(k)}$, is the fused LASSO estimator with $\xi = \tilde{\xi}^{(k)}$, $\zeta = 0$, and $M(\tilde{\xi}^{(k)})$ is the total number of non-zero estimates in $\tilde{\boldsymbol{g}}_1^{(k)} ... \tilde{\boldsymbol{g}}_J^{(k)}$. Let the value of ξ that optimizes the BIC in equation (3.2) be denoted as ξ^*. We then find the optimal value of ζ, over a grid of values, $\tilde{\zeta}^{(1)}, ..., \tilde{\zeta}^{(N)}$, starting with $\tilde{\zeta}^{(1)} = 0$ to $\tilde{\zeta}^{(N)} = 5$. We choose the value of ζ that maximizes the BIC,

$$\frac{1}{\hat{\sigma}^2} \sum_{j=1}^{J} \sum_{i=1}^{I} \left(y_{j,i} - \boldsymbol{z}_i' \tilde{\boldsymbol{g}}_j^{(n)} \right)^2 + log(I * J) M(\xi^*, \tilde{\zeta}^{(n)}), \tag{3.3}$$

where $\tilde{\boldsymbol{g}}_j^{(n)}$ is the FL estimate of $\boldsymbol{g}_j$ using penalty parameter values ξ^*, and $\tilde{\zeta}^{(n)}$, and $M(\xi^*, \tilde{\zeta}^{(n)})$ is the total number of non-fused groups in $\tilde{\boldsymbol{g}}_1^{(1)}, ..., \tilde{\boldsymbol{g}}_J^{(N)}$, which provides an unbiased estimate for the degrees of freedom under the FL (Tibshirani and Taylor, 2010).

3.2 Bayesian spatial approach for detection and identification

In this section we extend the BIMS method described in Section 2.4 to incorporate spatial information. We call the resulting method Spatial BIMS (SBIMS). In addition to placing mixtures of g-priors (Liang et al., 2008; Zellner, 1986) on the regression coefficients given the chemicals in the plume, Markov random fields (Besag, 1974) are used to define priors on the inclusion probabilities for each chemical. This prior captues our prior assumption that plumes are made of contiguous pixels. Full Bayesian estimation of our hierarchical model for the inclusion probabilities allows us to automatically account for multiplicities (Scott and Berger, 2006; Scott et al., 2010) while allowing for additional flexibility in modeling the actual concentration level of each chemical (when compared to that afforded by fussed LASSO penalties for the coefficients). On the other hand, the use of mixtures of g-priors for the regression coefficients allows us to more effectively deal with chemicals with highly correlated spectral signatures.

The problem of creating spatially dependent prior distributions for Bayesian model selection for a set of related regression models has been previously addressed using priors based on the Ising model (Ising, 1925), or its multi-class generalization, the Potts model (Potts, 1952); for example, see Li et al. (2012), Eches et al. (2011), and Zhang et al. (2014). However, joint estimation of the hyperparameters associated with these types of priors (which is key to ensure automatic multiplicity corrections) is challenging because of the lack of closed-form formulas for the normalizing constants of the priors. Instead, we impose spatial structure by using a clipped Gaussian Markov random field (cGMRF) (Berrett and Calder, 2012; De Oliveira, 2000; Weir and Pettitt, 2000), which dramatically simplifies the computational implementation of our hierarchical model.

Like the FL regression, cGMRF priors are defined in terms of a $J \times J$ symmetric neighborhood matrix, $\boldsymbol{\Upsilon}$ such that $\upsilon_{j,j'} = 1$ if and only if pixels j and j' are to be considered neighbors, and $\upsilon_{j,j'} = 0$ otherwise. As before, we concentrate on a first-order neighborhood matrix in which a neighborhood is composed of the horizontally and vertically adjacent pixels, but more general neighborhoods are possible. As the name suggests, the prior is defined hierarchically by "clipping" a Gaussian Markov random field. More specifically, for each chemical m we introduce a vector of (dependent) latent variables $\boldsymbol{\ell}_m = (\ell_{m,1}, \ldots, \ell_{m,J})$ such that

$$\ell_{m,j} \mid \ell_{m,1}, \cdots, \ell_{m,j-1}, \ell_{m,j+1}, \cdots, \ell_{m,J}, \alpha_m, \rho_m, u_m \sim \\ \mathsf{N}\left(\{1 - \rho_m\} \alpha_m + \rho_m \frac{\sum_{j' \neq j} \upsilon_{j,j'} \ell_{m,j}}{\sum_{j' \neq j} \upsilon_{j,j'}}, \frac{1}{u_m \sum_{j' \neq j} \upsilon_{j,j'}} \right), \quad (3.4)$$

and then let

$$\omega_{m,j}|\ell_{m,j} = \begin{cases} 1 & \text{if } \ell_{m,j} > 0, \\ 0 & \text{if } \ell_{m,j} \leq 0. \end{cases} \tag{3.5}$$

The prior in (3.4) and (3.5) incorporates spatial information by shrinking $\ell_{m,j}$ towards the value of its neighbors. The hyperparameters ρ_m and u_m control the level of spatial smoothing. In particular $\rho_m = 0$ implies that pixels are treated as independent a priori, and the spatial model reduces to an exchangeable model similar to the one discussed in Section 2.4. Futrhermore, by setting $0 < \rho_m < 1$ we ensure that the joint prior in (3.6) is proper, which is necessary to ensure that the posterior distribution of our model is also proper.

The cGMRF is a direct extension of the probit data augmentation scheme introduced in Albert and Chib (1993). Unlike a spatial probit generalized linear mixed model (e.g., see Banerjee et al., 2004), it imposes spatial structure directly on the latent variables. Indeed, note that the full conditional distributions defined by (3.4) imply a joint distribution of the form,

$$p(\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_M \mid a_m, u_m, \rho_m) = \prod_{m=1}^{M} \mathsf{N}\left(\alpha_m \mathbf{1}, \{u_m(\boldsymbol{D}_\Upsilon - \rho_m \boldsymbol{\Upsilon})\}^{-1}\right), \tag{3.6}$$

where $\boldsymbol{\ell}_m = (\ell_{m,1}, \ldots, \ell_{m,J})$ and $\boldsymbol{D}_\Upsilon = \operatorname{diag}\left\{\sum_{j\neq 1} \upsilon_{j,1}, \ldots, \sum_{j\neq J} \upsilon_{j,J}\right\}$. The use of this parsimonious structure is particularly important given the limited information available in a spatial binary process, especially in those that are not themselves directly observed (e.g., see Gelfand et al., 2000 and Berrett and Calder, 2012).

Under the cGMRF prior, the marginal prior probability that chemical m is present in pixel j is simply

$$\theta_{m,j} = \Phi\left(\frac{u_m^{1/2}\alpha_m}{q_{m,j,j}^{1/2}}\right),$$

where $\Phi(\cdot)$ denotes the cumulative distribution function of the standard normal and $q_{m,j,j}$ is the j-th diagonal element of the matrix $\mathbf{Q}_m = \{\boldsymbol{D}_\Upsilon - \rho_m \boldsymbol{\Upsilon}\}^{-1}$. As in Section 2.4, we are interested in allowing the model to automatically adjust for multiplicities by making these probabilities random. Hence, we assign α_m, ρ_m and u_m prior distributions of the form

$$\alpha_m \sim \mathsf{N}\left(\alpha_0, \phi^2\right), \quad u_m \sim \mathsf{Gam}\left(\psi_1, \psi_2\right), \quad \rho_m \sim \mathsf{Uni}\{0.5, 0.75, 0.9, 0.95, 0.975, 0.99, 0.999\}.$$

To match the approach discussed in Section 2.4 we select the hyperparameters of the prior to ensure that $\theta_{m,j}$ are approximately uniform. In particular, the hyperparameters are chosen so that

$$\eta_m = \frac{u_m^{1/2} \alpha_m}{\left(\frac{1}{J}\sum_{j=1}^{J} q_{m,j,j}\right)^{1/2}}$$

has mean 0 and variance 1. For first-order neighborhoods, this leads to $\alpha_0 = 0$, $\phi = 1$, $\psi_1 = 2$ and $\psi_2 = 8$. Alternatively, if prior information about the likelihood of a specific chemical is available, it can be incorporated through different specifications of the hyperparameters (e.g., we could use a smaller mean for α_m if we believe that chemical m is unlikely to be present in the sample).

Having an approximately uniform prior on the chemical inclusion probabilities provides one constraint for the choices of the hyperparameters, α_0, ϕ, ψ_1, and ψ_2 in the priors for α_m and u_m. We chose $\alpha_0 = 0$ and $\phi = 1$ to match the probit scale of the α_m parameters. Prior sensitivity analysis was performed by considering different values of ψ_1 and ψ_2, and although performance did not change much (even for different scenarios) we found that $\psi_1 = 2$, and $\psi_2 = 8$ yielded good empirical results. The prior choice on ρ_m did not undergo prior sensitivity analysis because it was motivated by Gelfand and Vounatsou (2003). They show that $\rho_m < 1$ is necessary for propriety, and that $\rho_m > 0$ is necessary for inducing spatial dependence. Furthermore, they show that values of ρ_m near 1 are needed for even moderate spatial dependence, which is why the uniform prior on ρ_m that we use has more values closer to one.

3.3 Parallel Implementation for Spatial Methods

Since realistic applications of our work involve a large number of pixels (∼80,000) and spectral bands (∼100) we implemented our methods in a parallel environment using `openMP`. For the pixel-by-pixel methods, like the LASSO and BIMS that assume independence over all pixels, an embarrassingly parallel algorithm can be used by distributing computation over all pixels simultaneously. As for the spatial methods, specifically the FL, and SBIMS (and spatial non-parametric BIMS, which we introduce in Chapter 4), these algorithms can be parallelized using a "checkerboard" approach (e.g., see Landau, 2005), which greatly reduces processing time. To understand how this scheme works, consider labeling the pixels in the image as "red" or "black" in an alternating pattern similar to that on a checkerboard (see Figure 3.1). Under a first-order neighborhood structure, pixels will either be labelled as “red” or “black”. The “red” pixels are conditionally independent from each other given those of the corresponding “black” pixels, and vice versa. Hence, an embarrassingly parallel implementation is again possible by distributing computation for the “red” pixels first, and once these have been updated, distributing computation for the “black” pixels. A similar approach can be employed for higher order neighborhood structures by partitioning the pixels into a larger number of subgroups/colors.

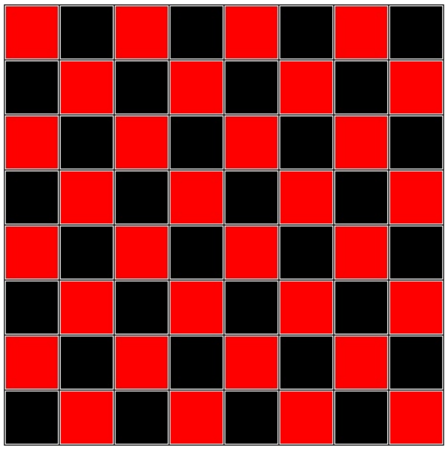


Figure 3.1: For spatial methods, we can consider labeling a hyperspectral image like checkerboard where each pixel is either “red” or “black”. We can see that “red” pixels are conditionally independent given the “black” pixels, and vice versa. Therefore we can first update “red” pixels in parallel holding the “black” pixels fixed, then we can update the “black” pixels in parallel holding the “red” pixels fixed.

3.4 Evaluation of FL and SBIMS on closely linear dataset

In this section we evaluate the performance of the FL and SBIMS on the closely linear dataset described in Section 1.5. As in Section 2.5, we assume that TEP, Sarin, and DMMP are the only possible chemicals in the image. Figure 3.2 shows the confusion matrices for the FL and SBIMS. Unlike the performance of the LASSO and BIMS discussed in Section 2.5, neither the FL nor SBIMS makes false positives in the sense of identifying pixels with no gas as plume present. The FL makes no false negatives, in the sense of identifying plume present pixels as having no gas, whereas SBIMS only makes 2 false negative mistakes. Hence, these spatial methods are excellent detection methods in that the can easily distinguish between plume present pixels and no plume present pixels. However, it is clear that SBIMS is the superior identification method in terms of classifying most pixels into their correct chemical class. We can see that the FL identifies all pixels in the "TEP/DMMP" class incorrectly as having all gasses, and slightly struggles classifying some of the pixels in the "All" class. As mentioned previously, SBIMS makes only 2 false negatives, and impressively, those are the only mistakes SBIMS makes at all.

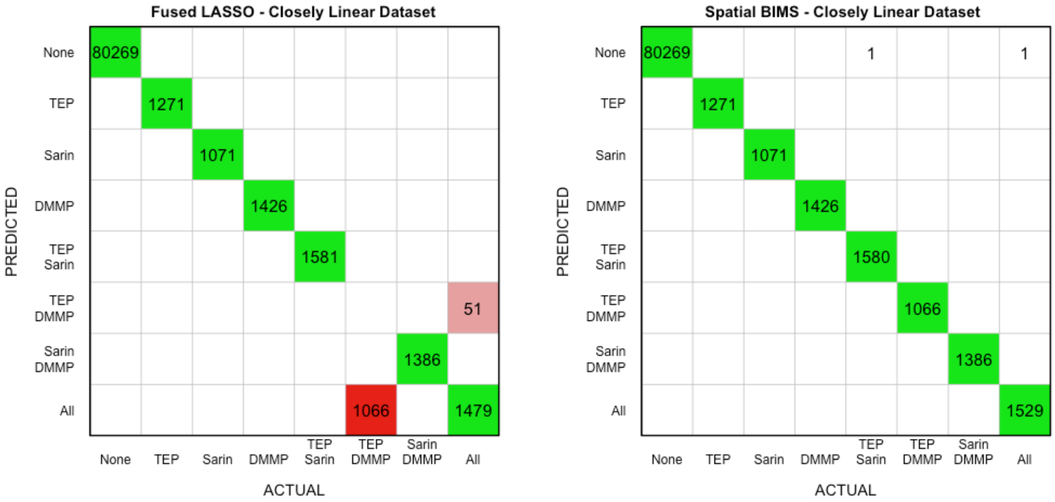


Figure 3.2: Confusion matrices for the FL and SBIMS.

	LASSO	BIMS	FL	SBIMS
ACC	0.9311	0.9972	0.9875	0.9998
MCC	0.8654	0.9859	0.9369	0.9999

Table 3.1: ACC and MCC values for the simulated dataset.

In Table 3.1 we provide the ACC and MCC values of the FL and SBIMS, plus those of the LASSO and BIMS for reference. We can see that SBIMS has higher ACC and MCC values than all other methods. Since the ACC looks at only diagonal elements in a confusion matrix while the MCC also looks at off-diagonal elements (see section 1.6), it is typically harder for a given method to attain an MCC value that is higher than its ACC value. However, we can see that in fact the SBIMS method does attain a higher MCC value attesting to its superior performance. We can also see that the non-spatial method, BIMS, has higher ACC and MCC values than the FL, even though FL takes into account spatial information. Figure 3.3 paints a more nuanced picture. We see that the FL no longer has the random mistakes that the BIMS (and LASSO) make, and therefore in that sense, one could argue that the FL is still a better detector (i.e., detecting gas from no-gas pixels) than the BIMS. Lastly, Figure 3.3 clearly shows the smoothing

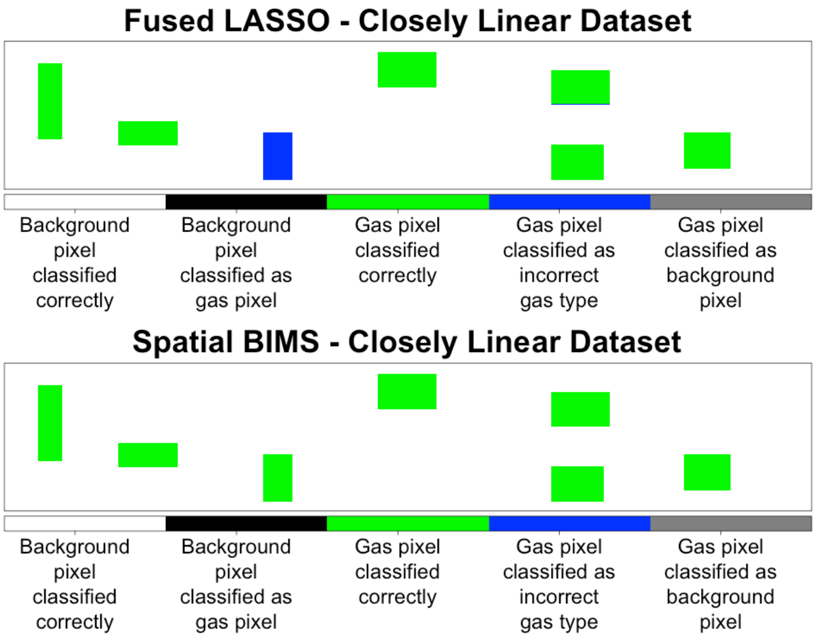


Figure 3.3: Identification results mapped onto the area covered by the synthesized data.

effects that the FL and SBIMS have on identification results.

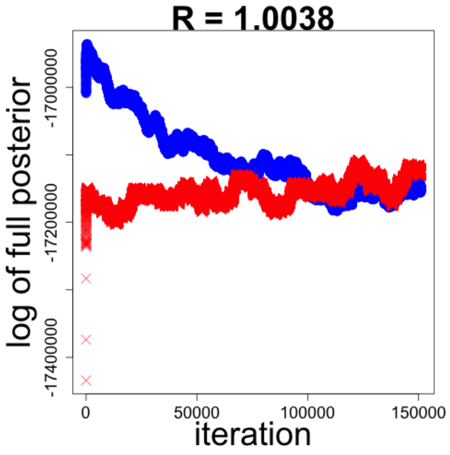


Figure 3.4: Trace plot of log posterior distribution under SBIMS for the closely linear dataset for two chains, where one chain is represented by red x's, and the second chain is represented by blue dots. Using all 150,000 iterations in each of the two chains a Gelman and Rubin statistic of 1.0038 was calculated implying convergence was reached.

For both the FL and SBIMS on the closely linear dataset we implemented the checkerboard scheme described in section 3.3. The FL took 4 days to solve. Recall from section 3.1 that we solve the FL over a grid of LASSO penalties and a grid of FL LASSO penalties to find the optimal common LASSO penalty and optimal FL LASSO penalty using BIC. Because the FL is a much more complex optimization problem than the LASSO, for one pair of penalties it takes around 2 hours for the FL to solve. The Split Bregman algorithm (see Chapter 5) used to solve the FL requires specification of certain parameters that could be fine tuned to increase the FL's run time efficiency.

Like with BIMS, SBIMS requires an MCMC algorithm to obtain posterior samples. SBIMS on the closely linear dataset converged after 150,000 iterations which took 3.125 days or 1.8 seconds per iteration. We then ran the MCMC for 10,000 more iterations which we used for inference. The time per iteration for SBIMS is not drastically greater than the time per iteration for BIMS. However the total time for SBIMS is much greater than BIMS because SBIMS requires much more iterations to reach convergence due to greater complexity. We again verified MCMC convergence by running two MCMC chains and we show the log posterior distribution values for each in Figure 3.4. Using all 150,000 iterations from the two chains, we obtained a Gelman and Rubin diagnostic value of 1.0038, which is less than 1.2, and therefore implies convergence was reached.

3.5 Evaluation of methods on realistic dataset

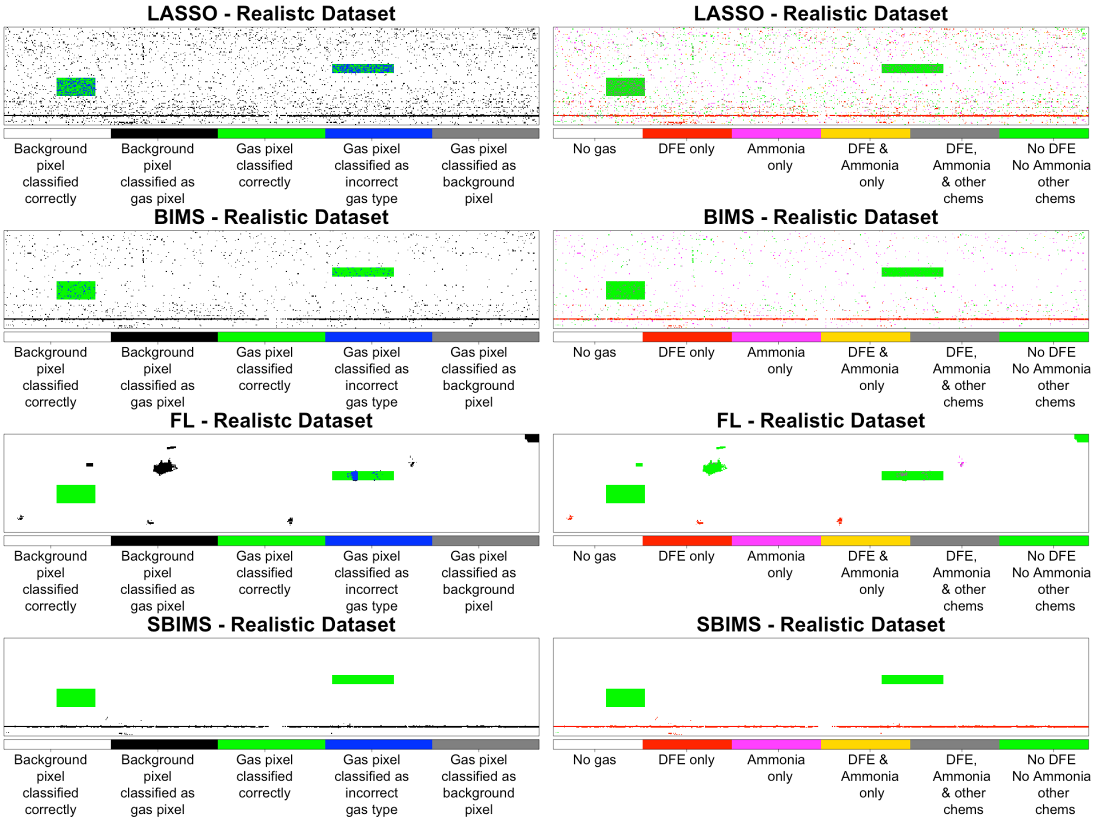
We now evaluate all methods, the LASSO, BIMS, FL, and SBIMS, on the realistic dataset described in Section 1.5. Figure 3.5 shows identification results for the four identification methods, and Table 3.2 displays confusion matrix metrics. From the top two left plots in Figure 3.5 we can see that in general the LASSO and BIMS perform worse than on the closely linear dataset mainly because we are considering two additional chemicals (DFE and Ammonia) that we did not consider for the closely linear dataset. We see a lot more speckled mistakes under both methods. However, like the closely linear dataset the BIMS method is outperforming the LASSO for no gas pixels as well as for plume pixels. One reason for BIMS' better performance can be attributed to the fact that it takes into account colinearity among chemical signatures.

Like with the closely linear dataset, we can see in Figure 3.5 that both the FL and SBIMS have spatial smoothing effects. However, apart from a line object, which we describe more in the next paragraph, SBIMS is better able to smooth out no-gas pixels than FL, and it is especially better at smoothing out plume pixels than the FL. Interestingly looking at Table 3.2 the FL yields higher ACC and MCC values than SBIMS, and this is mainly because SBIMS is detecting a line object that the FL is not. However, based on the results of the LASSO and BIMS we argue that SBIMS is providing results that are closer to the truth. Of course, SBIMS and FLASSO are different in several ways, however a few reasons that SBIMS is performing better than the FL is because like BIMS, it takes into account colinearity among chemicals, and secondly, it places spatial dependence on the prior chemical inclusion probabilities whereas the FL places spatial dependence on chemical coefficients.

Something that is immediately apparent is that there is a line object being detected by the LASSO, BIMS, and SBIMS in an area of the image in which we did not embed any plumes. The reason for this is because of the particular simulation setting, specifically the possible chemicals that we are considering. Recall from subsection 1.5.1 that for the closely linear dataset we embed plumes with combinations of TEP, Sarin, and DMMP, and we consider only TEP, Sarin, and DMMP as possible chemicals that could be present in the closely linear image. For the realistic dataset we also embed plumes with combinations of TEP, Sarin, and DMMP, but we consider TEP, Sarin, DMMP, plus DFE, and ammonia as possible chemicals in the realistic image, where DFE and ammonia were not embedded. The right plots of Figure 3.5 (except for the FL plot) show that it is mainly DFE that is being identified in this line object. Therefore, the spectral composition of these pixels are naturally similar to the spectral composition of DFE. Since we did not include DFE as a possible chemical when analyzing the closely linear dataset, we did not detect the line object in Figures 2.3 and 3.3.

Interestingly, the FL does not detect the line object, as shown in third pair of plots down in Figure 3.5. Recall that the FL is a spatial penalization method that can shrink coefficients to zero and produce a spatial effect by shrinking the difference between neighboring pixels. Of course, SBIMS is a also a spatial method that smooths neighboring pixels to be similar and can indirectly through the use of the indicator variables, $\omega_{m,j}$, shrink coefficients to zero. However, the reason why the FL is not detecting the line object, whereas SBIMS does, is because since the FL places spatial dependence directly on regression coefficients, it tends to over smooth. SBIMS on the other hand places spatial dependence on chemical inclusion probabilities. We will again see FL's tendency to over smooth when we analyze the challenge dataset in Section 3.6.

To dive deeper into the results of the realistic dataset, we now discuss a series of tables meant to convey similar information in a confusion matrix. Note that since we are considering five possible chemicals it is hard to present a confusion matrix in its traditional form. Table 3.3



LASSO - Realistc Dataset

LASSO - Realistic Dataset

BIMS - Realistic Dataset

BIMS - Realistic Dataset

FL - Realistc Dataset

FL - Realistic Dataset

SBIMS - Realistic Dataset

SBIMS - Realistic Dataset

Figure 3.5: Left plots show the identification results for the LASSO, BIMS, FL, and SBIMS. Right plots explain that the line detected across the bottom of the image is identifying DFE, a chemical we did not include as a possible chemical for the closely linear dataset. However FL does not detect the line object because it has a tendency to over smooth.

	LASSO	**BIMS**	**FLASSO**	**SBIMS**
ACC	0.9304	0.9721	0.9903	0.9864
MCC	0.4138	0.6636	0.8370	0.7998

Table 3.2: ACC and MCC values for the realistic dataset.

tallies the number of no-gas pixels that identified zero chemicals, 1 chemical, 2 chemicals, etc over the four methods. The last row of Table 3.3 shows that there were a total of 87, 404 pixels that had no gas. In general we can see that the vast majority of no-gas pixels were identified correctly as having no chemicals. We can then see that the number of pixels tallied decreases as the number of identified chemicals increases for all methods, with the exception of BIMS which identifies more pixels as having 5 chemicals than 4 chemicals. For the no gas pixels it looks like SBIMS is the closest to the truth because it only identifies no gas pixels as having 0 chemicals or 1 chemical.

Number of chemicals identified for no gas pixels						
	0	**1**	**2**	**3**	**4**	**5**
LASSO	82118	4120	892	194	71	9
BIMS	85075	2165	130	20	4	10
FL	86669	535	184	11	5	0
SBIMS	86190	1214	0	0	0	0
Truth	87404	0	0	0	0	0

Table 3.3: Tally of pixels by number of chemicals identified for no gas pixels in realistic dataset. Note that each row should sum up to the same number of pixels.

Table 3.4 is similar to Table 3.3 except that it tallies pixels in the TEP/Sarin plume (left outlined plume in Figure 1.8, which corresponds to the left rectangular green regions in plots of Figure 3.5) by the number of chemicals identified. We can see that for all methods either two or more chemicals were identified in TEP/Sarin pixels, which is a positive result since the truth is that all pixels in the TEP/Sarin plume have 2 chemicals. For the LASSO we again see that the number of tallied pixels decreases as the number of identified chemicals increases. The BIMS on the other hand identifies one more pixel as having 5 chemicals than four chemicals. The FL and SBIMS were able to identify the correct number of chemicals for all TEP/Sarin pixels.

Number of chemicals identified for TEP/Sarin pixels						
	0	**1**	**2**	**3**	**4**	**5**
LASSO	0	0	788	324	101	11
BIMS	0	0	1119	86	9	10
FL	0	0	1224	0	0	0
SBIMS	0	0	1224	0	0	0
Truth	0	0	1224	0	0	0

Table 3.4: Tally of pixels by number of chemicals identified for TEP/Sarin pixels in realistic dataset. Note that each row should sum up to the same number of pixels.

Table 3.5 is similar to tables 3.3 and 3.4 except that it tallies pixels in the TEP/DMMP plume (right outlined plume in Figure 1.8, which corresponds to the right rectangular green regions in plots of Figure 3.5) by the number of chemicals identified. We again see that for all methods either two or more chemicals were identified in TEP/DMMP pixels. And both the LASSO and BIMS have similar behavior to the TEP/Sarin results as the number of identified chemicals increases. However, unlike the TEP/Sarin pixels, the FL mistakenly identifies some pixels as having three or four chemicals. SBIMS on the other hand identifies all TEP/DMMP pixels correctly. One reason the FL performs perfectly for the TEP/Sarin plume, but not for the TEP/DMMP plume is because the FL does not take into account colinearity among chemical signatures, and TEP and DMMP are highly correlated, whereas TEP and Sarin are less correlated.

Number of chemicals identified for TEP/DMMP pixels						
	0	**1**	**2**	**3**	**4**	**5**
LASSO	0	0	461	403	86	22
BIMS	0	0	903	56	6	7
FL	0	0	842	103	27	0
SBIMS	0	0	972	0	0	0
Truth	0	0	972	0	0	0

Table 3.5: Tally of pixels by number of chemicals identified for TEP/DMMP pixels in realistic dataset. Note that each row should sum up to the same number of pixels.

Table 3.6 shows the number of no gas pixels in which no chemicals were identified, TEP was identified, Sarin was identified, etc. We see that DFE was incorrectly identified the most for all methods except the FL because of the line object. However, apart from that there does not seem to be a clear pattern regarding which chemicals are being incorrectly identified in no-gas pixels. We do see that SBIMS does not identify TEP, sarin, or ammonia in any of the no-gas pixels. Furthermore, with the exception of 3 pixels incorrectly identified as having DMMP, the SBIMS method is identifying no-gas pixels as having either no gas, or DFE. Furthermore, looking at Figure 3.5 the no-gas pixels that SBIMS incorrectly identifies as having DFE are mainly located in the line object.

Chemicals identified for no gas pixels						
	None	**TEP**	**Sarin**	**DMMP**	**DFE**	**Ammonia**
LASSO	82118	301	1154	858	2704	1762
BIMS	85075	34	107	316	1297	797
FL	86669	311	255	270	90	30
SBIMS	86190	0	0	3	1211	0
Truth	87404	0	0	0	0	0

Table 3.6: Tally of pixels by identified chemcials for no gas pixels in realistic dataset. Note that the classes in each column are not mutually exclusive and therefore each row does not necessarily need to add up to the same number of pixels.

The left side of Table 3.7 shows that for all methods, TEP and Sarin was identified in all pixels in the TEP/Sarin plume, but for some pixels additional chemicals to TEP and Sarin were also identified. The BIMS is able to identify significantly more TEP/Sarin pixels correctly than the LASSO, whereas both the FL and SBIMS methods perform perfectly. The right side of Table 3.7 shows information for only the TEP/Sarin pixels that were identified as having TEP, Sarin, and other chemicals. Specifically the information shown is the pixel count for DMMP, DFE, and Ammonia. We see that all three chemicals are incorrectly showing up for the LASSO and BIMS, and no other clear pattern emerges.

	Chemicals identified for TEP/Sarin pixels	
	TEP, Sarin only	**TEP, Sarin, +other chems**
LASSO	788	436
BIMS	1119	105
FL	1224	0
SBIMS	1224	0
Truth	1224	0

	Chemicals identified for TEP, Sarin, +other chem pixels		
	DMMP	**DFE**	**Ammonia**
LASSO	126	241	192
BIMS	17	37	80
FL	0	0	0
SBIMS	0	0	0
Truth	0	0	0

Table 3.7: Left table shows that for all methods pixels in the TEP/Sarin plume (left plume) were labeled as having only TEP/Sarin or having TEP/Sarin and other chemicals. Right table tallies the number of pixels by the additional chemical identified for pixels in the TEP, Sarin + other chems class from the left table. Rows in the left side should add up to the same number of pixels, whereas rows on right side do not need to add up to the same number of pixels.

Table 3.8 is similar to Table 3.7 except it is for the TEP/DMMP plume. Like the TEP/Sarin plume, we again see that all methods are able to identify TEP and DMMP in the TEP/DMMP plume, but some additional chemicals are being identified in some of the pixels. Interestingly we see that the BIMS outperforms the FL for TEP/DMMP pixels, and the SBIMS method performs perfectly. Also the LASSO identifies more pixels as having additional chemicals than having only TEP and DMMP present. Again, one reason for the comparatively better performance of the Bayesian approaches is because they consider correlation among chemical signatures. The right side of Table 3.7 shows the number of pixels that identified Sarin, DFE, and Ammonia for the TEP/DMMP pixels that were identified as having additional chemicals to TEP and DMMP. We can see that all additional chemicals are being incorrectly identified in TEP/DMMP pixels for the LASSO, BIMS, and FL.

The LASSO was solved over 194 LASSO penalties from 0 to 19.3 and took 8.7 minutes. We solved the FL over 26 LASSO and FL penalty pairs, which took 3 days to complete. For BIMS we ran one MCMC chain of 10,000 iterations with a burn in of 5,000, and for SBIMS we ran one MCMC chain for 160,000 iterations with a burn in of 150,000. Although we do not run two

	Chemicals identified for TEP/DMMP pixels			Chemicals identified for TEP, DMMP, +other chem pixels		
	TEP, DMMP only	TEP, DMMP, +other chems		Sarin	DFE	Ammonia
LASSO	461	511	**LASSO**	409	130	102
BIMS	903	69	**BIMS**	10	23	56
FL	842	130	**FL**	85	49	23
SBIMS	972	0	**SBIMS**	0	0	0
Truth	972	0	**Truth**	0	0	0

Table 3.8: Left table shows that for all methods pixels in the TEP/DMMP plume (right plume) were labeled as having only TEP/DMMP or having TEP/DMMP and other chemicals. Right table tallies the number of pixels by the additional chemical identified for pixels in the TEP, DMMP + other chems class from the left table. Rows in the left side should add up to the same number of pixels, whereas rows on right side do not need to add up to the same number of pixels.

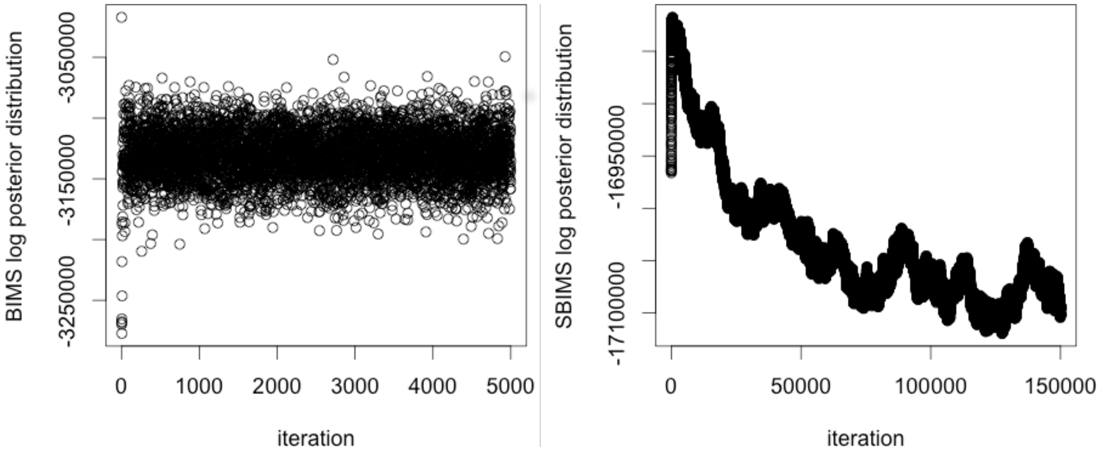


Figure 3.6: Trace plot of posterior distribution under BIMS (left plot) and SBIMS (right plot) for the realistic dataset.

MCMC chains for the Bayesian methods, we ran each method for the same number of iterations that reached convergence under the closely linear dataset. Total computation time for BIMS was 3 hours or 1.2 seconds per iteration, and SBIMS took 3.6 days in total or 2.1 seconds per iteration. We show log posterior values under BIMS for the last 5,000 iterations, and for 150,000 iterations under SBIMS in Figure 3.6. Note that BIMS and SBIMS run times for the realistic dataset are longer than those for the closely linear dataset because we are considering 5 possible chemicals in the realistic dataset, but only 3 possible chemicals for the closely linear dataset.

3.6 Evaluation of methods on the MIT Lincoln Lab Challenge Data

In this section we evaluate all four methods (LASSO, BIMS, FL, and SBIMS) on the Challenge dataset described in Section 1.5.2. It is important to note that in this evaluation we consider eight possible chemicals that could be present in the image, as opposed to the three for the closely linear dataset, and as opposed to five in the realistic dataset. Therefore, this will increase the level of detection and identification difficultly.

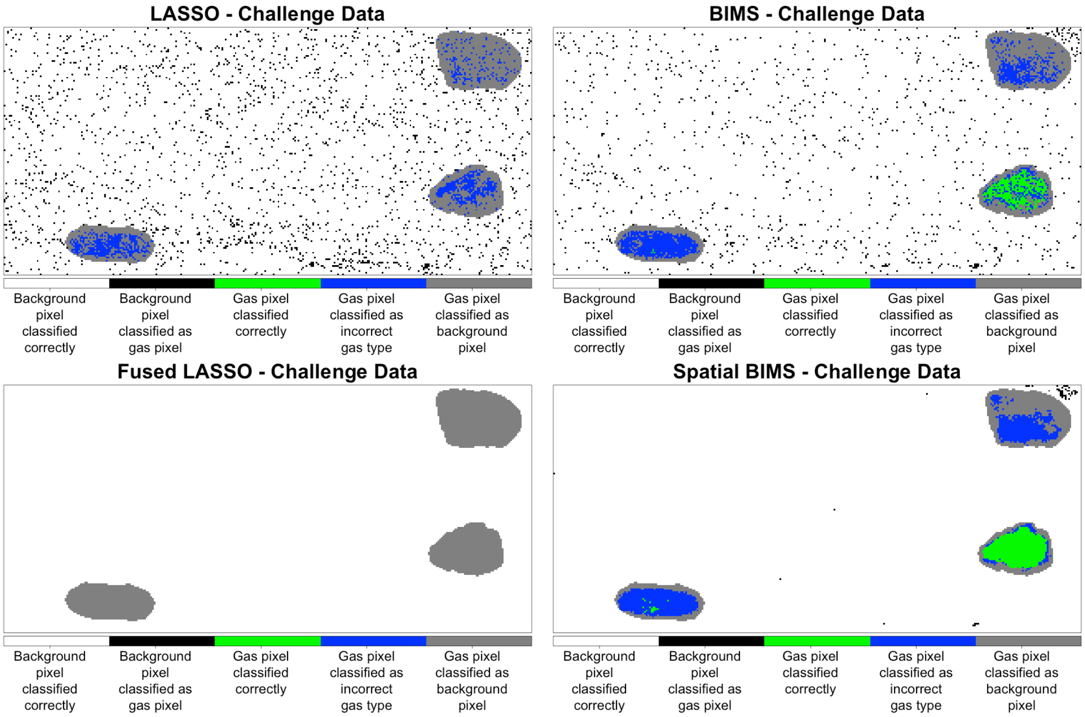


Figure 3.7: Identification results for all four classification methods mapped onto the area covered by the MIT Lincoln Lab challenge data.

	LASSO	BIMS	FLASSO	SBIMS
ACC	0.8817	0.9101	0.9239	0.9370
MCC	0.1070	0.2941	0.0000	0.4747

Table 3.9: ACC and MCC values for the MIT Lincoln Lab challenge data.

Figure 3.7 presents the results of the four models mapped onto the area covered by the MIT Lincoln Lab challenge data, while Table 3.9 presents the summary metrics of the associated confusion matrices. Relative to the semi-synthetic data in Sections 2.5, 3.4, and 3.5 all methods demonstrate lower accuracy for the challenge data cube. In particular, we see a large number of misclassified background pixels under BIMS and LASSO. However we do see that the BIMS outperforms the LASSO as it makes less mistakes (note the higher ACC and MCC values), and it is somewhat better at detecting the presence of plumes 2 and 3 and identifying the chemicals in cloud 1. In fact, the LASSO is unable to identify any pixels correctly in plume 1. We can also see that the inclusion of spatial information in the Bayesian setting substantially improves the results. Indeed, under SBIMS the number of false positives associated with background pixels decreases substantially, and gas identification improves substantially particular in plume 1 (and to a lesser degree, on plume 3). On the other hand, FLASSO has a very poor performance in this example as it over-smooths and is therefore unable to detect any of the plumes. Lastly we highlight the gray ring of misclassified plume pixels as background pixels that is very clear in the BIMS and SBIMS results especially for plume 1 (bottom right plume) and plume 3(bottom left plume). Since we did not embed the plumes ourselves we do not know the value of the embedding parameters (i.e., chemical concentrations, plume temperatures, embedding masks, etc.) that were inputted into the embedding algorithm. However, due to the gray rings we hypothesize that a Gaussian embedding mask was used for the embedded plumes (see Section 1.3).

Presenting a confusion matrix for the challenge dataset is especially difficult because it would have dimension 256×256. Therefore, like the realistic dataset we present tables to convey similar information that would be presented in a confusion matrix. In Table 3.10 we tally the number of pixels that identified zero chemicals, 1 chemical, 2 chemicals, and up to 8 chemicals, which is the total number of possible chemicals considered for the challenge dataset. In general we do see that the number of pixels decreases as the number of identified chemicals increases, with the exception of the LASSO which classifies more 8-chemical pixels than 6- and 7-chemical pixels. Interestingly no method identifies 7 chemicals for the no gas pixels, and this is true for the plume pixels as shown in Tables 3.11 - 3.13. We also see that the majority of no-gas pixels are being identified as having zero chemicals, with FL identifying all no gas pixels correctly. However, we know from Figure 3.7, that the FL identifies all pixels as having no gas. SBIMS is able to classify almost all no-gas pixels correctly except for 50 no gas pixels which are identified as having only 1 chemical present.

Number of chemicals identified for no gas pixels									
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
LASSO	41756	2195	311	44	12	15	3	0	11
BIMS	43320	891	100	17	8	1	0	0	10
FL	44347	0	0	0	0	0	0	0	0
SBIMS	44297	50	0	0	0	0	0	0	0
Truth	44347	0	0	0	0	0	0	0	0

Table 3.10: Tally of pixels by number of chemicals identified for no gas pixels in challenge dataset. Note that each row should sum up to the same number of pixels.

We show the same information in Table 3.11 as in Table 3.10 except for the pixels in plume 1 (bottom right plume). We can immediately see that the LASSO does a poor job of detecting that a plume is present in the plume 1 region. This result is unlike the results for the closely linear dataset and especially the realistic dataset (see Table 3.4) where LASSO (and all other methods) does a good job of detecting that gas is present in plume regions. BIMS performs better than the LASSO in detecting where gas is present, but it still detects more pixels as having no gas than having at least 1 chemical present. SBIMS on the other hand does the best job in identifying the correct number of chemicals present in plume 1.

Number of chemicals identified for plume 1 pixels									
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
LASSO	739	44	8	9	12	19	42	0	229
BIMS	499	127	407	50	13	2	0	0	4
FL	1102	0	0	0	0	0	0	0	0
SBIMS	348	95	659	0	0	0	0	0	0
Truth	0	0	1102	0	0	0	0	0	0

Table 3.11: Tally of pixels by number of chemicals identified for plume 1 pixels in challenge dataset. Note that each row should sum up to the same number of pixels

Interestingly the LASSO does a better job in identifying the correct number of chemicals for pixels in plume 2 (top right plume) than for the pixels in plume 1 (bottom right plume) as shown in Table 3.12. BIMS on the other hand is performing worse in terms of identifying the correct number of chemicals in plume 2 pixels than in plume 1 pixels. And SBIMS is not able to identify any plume 2 pixels as having the correct number of chemicals. SBIMS is identifying either 0 chemicals or 1 chemical for the plume 2 pixels. In comparison to plume 1 and plume 3 (see Table 3.13) all methods are poor detectors as they detect more plume 2 pixels as having no gas than having at least 1 chemical.

Number of chemicals identified for plume 2 pixels									
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
LASSO	1297	139	51	20	5	16	3	0	2
BIMS	1149	355	22	6	1	0	0	0	0
FL	1533	0	0	0	0	0	0	0	0
SBIMS	982	551	0	0	0	0	0	0	0
Truth	0	0	1533	0	0	0	0	0	0

Table 3.12: Tally of pixels by number of chemicals identified for plume 2 pixels in challenge dataset. Note that each row should sum up to the same number of pixels

Although all methods struggle to identify the correct number of chemicals for plume 3 pixels as shown in Table 3.13, they are better at detecting gas present pixels in plume 3 (bottom left plume) than in plume 2. This is especially true for the BIMS and SBIMS methods which identify more plume 3 pixels as having at least 1 chemical than having zero chemicals.

Number of chemicals identified for plume 3 pixels									
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
LASSO	558	158	82	29	43	75	36	0	27
BIMS	441	534	35	4	3	0	0	0	1
FL	1018	0	0	0	0	0	0	0	0
SBIMS	344	654	20	0	0	0	0	0	0
Truth	0	0	1018	0	0	0	0	0	0

Table 3.13: Tally of pixels by number of chemicals identified for plume 3 pixels in challenge dataset. Note that each row should sum up to the same number of pixels

In Table 3.14 we tally the number of no gas pixels that identified no chemicals, chemical 1, chemical 2, etc over the four methods. For the LASSO there are some chemicals that are being incorrectly identified more than other chemicals. However, for BIMS the number of pixels identified in each of the chemical classes seems to be more uniform. The fact that there is not really a clear pattern for the incorrectly classified gas pixels under the LASSO and BIMS is evidence that these methods are making random mistakes because they assume all pixels are independent. SBIMS on the other hand is identifying almost all no-gas pixels correctly, and only identifies chemicals, 1, 2, and 4 in incorrectly classified no gas pixels.

In Table 3.15 we tally the number of pixels over chemical classes of interest for plume 1. Only the Bayesian methods (BIMS and SBIMS) are able to correctly classify pixels in plume 1. BIMS is able to classify about 34% of plume 1 pixels correctly, whereas SBIMS is able to correctly identify about 60% of plume 1 pixels correctly. Interestingly for the plume 1 pixels in

Chemicals identified for no gas pixels									
	C0	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**
LASSO	41756	33	1347	11	349	11	375	1027	25
BIMS	43320	91	267	145	264	133	90	156	113
FL	44347	0	0	0	0	0	0	0	0
SBIMS	44297	45	1	0	4	0	0	0	0
Truth	44347	0	0	0	0	0	0	0	0

Table 3.14: Tally of pixels by identified chemicals for no gas pixels in challenge dataset. Since classes in columns are not mutually exclusive, it is not necessary that all rows add up to the same number of pixels.

which SBIMS detected gas, only chemicals C1 and C3 were identified, i.e. no other extraneous chemicals are identified under SBIMS. However for the LASSO and BIMS methods chemicals other than C1 and C3 are being identified. We did look closer into which additional chemicals are being identified for the LASSO and BIMS, however no clear pattern emerged except for the fact that each extraneous chemical, namely C2, C4, C5, C6, C7, and C8, were identified in at least one pixel for plume 1 under the LASSO and BIMS.

Chemicals identified for plume 1 pixels that contains C1 and C3								
	None	**C1 only**	**C1, no C3 + other chems**	**C3 only**	**C3, no C1 + other chems**	**C1, C3 only**	**C1, C3, + other chems**	**Chems other than C1, C3**
LASSO	739	1	26	0	8	0	277	51
BIMS	499	33	2	80	42	376	56	14
FL	1102	0	0	0	0	0	0	0
SBIMS	348	17	0	78	0	659	0	0
Truth	0	0	0	0	0	1102	0	0

Table 3.15: Tally of pixels by identified chemicals class for plume 1 (bottom right plume) pixels in challenge dataset. Note that each row should sum up to the same number of pixels

Only the BIMS method is able to identify one pixel correctly in plume 2 as shown in Table 3.16. The SBIMS method identifies plume 2 pixels as having either no gas or having only chemical C6 present, even though the truth is that both chemicals C6 and C5 are present. Similar to plume 1, the SBIMS method is not identifying any other additional chemicals to chemical C6 (and C5). The LASSO and BIMS on the other hand are identifying extraneous chemicals, and we also did not find a clear pattern of which extraneous chemicals are being identified, except that each chemical other than C5 and C6 are being identified in some of the plume 2 pixels.

Chemicals identified for plume 2 pixels that contains C5 and C6								
	None	**C5 only**	**C5, no C6 + other chems**	**C6 only**	**C6, no C5 + other chems**	**C5, C6 only**	**C5, C6, + other chems**	**Chems other than C5, C6**
LASSO	1297	0	0	99	86	0	2	49
BIMS	1149	106	5	183	12	1	0	77
FL	1533	0	0	0	0	0	0	0
SBIMS	982	0	0	551	0	0	0	0
Truth	0	0	0	0	0	1533	0	0

Table 3.16: Tally of pixels by identified chemicals class for plume 2 (top right plume) pixels in challenge dataset. Note that each row should sum up to the same number of pixels

For plume 3 we can see that only BIMS and SBIMS can correctly identify pixels, but for a very low number of pixels as shown in Table 3.17. SBIMS only identifies chemicals that are actually present in plume 3, specifically chemicals C7 and C8. LASSO and BIMS on the other hand are identifying chemicals that are not actually present in plume 3. As with plumes 1 and 2, there was no evident pattern among extraneous chemicals being identified, except that all chemicals were identified over the plume 3 pixels under the LASSO and BIMS.

Chemicals identified for plume 3 pixels that contains C7 and C8								
	None	**C7 only**	**C7, no C8 + other chems**	**C8 only**	**C8, no C7 + other chems**	**C7, C8 only**	**C7, C8, + other chems**	**Chems other than C7, C8**
LASSO	558	154	112	0	23	0	157	14
BIMS	441	202	17	327	15	4	3	9
FL	1018	0	0	0	0	0	0	0
SBIMS	344	18	0	636	0	20	0	0
Truth	0	0	0	0	0	1018	0	0

Table 3.17: Tally of pixels by identified chemicals class for plume 2 pixels in challenge dataset. Note that each row should sum up to the same number of pixels

We evaluated the LASSO in 55 minutes over 868 LASSO penalty values from 0 to 0.0867 at which all coefficients were shrunk to zero. Recall for the closely linear dataset that it took a LASSO penalty value of 5.8 to shrink all coefficients to zero. This suggests that the coefficients for the challenge dataset are much smaller than those for the closely linear dataset. The FL was evaluated over 121 LASSO and FL penalty pairs, which took 12 hours to complete. The FL takes less time for the challenge dataset than the closely linear and realistic dataset due to

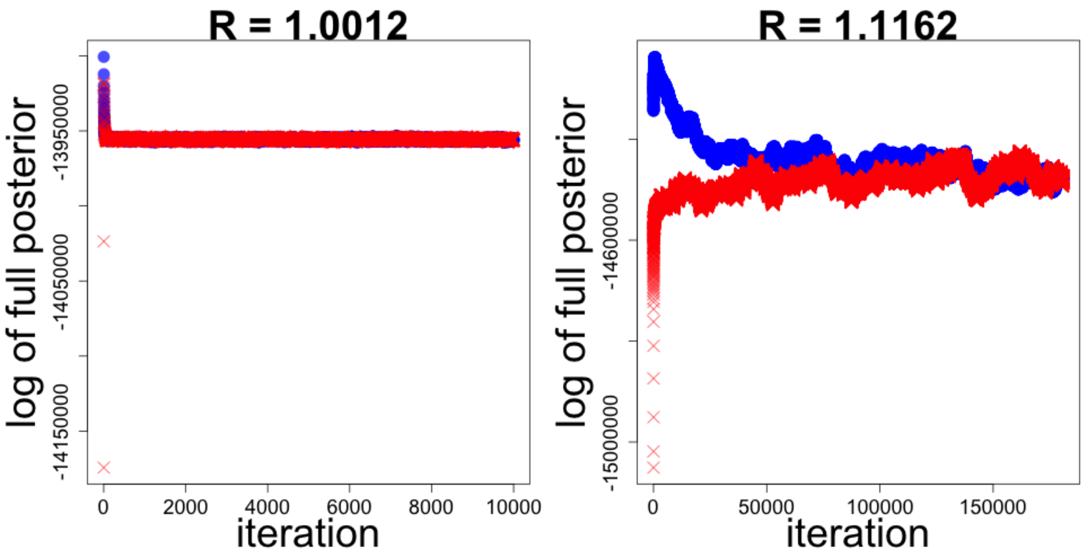


Figure 3.8: Trace plots of log posterior distribution under BIMS (left plot) and SBIMS (right plot) for the challenge dataset. Under each method 2 chains were run, where one chain is represented by red x's, and the second chain is represented by blue dot's. Using the first 5,000 iterations in the 2 chains under both BIMS, and all 180,000 iterations in the 2 chains under SBIMS, both methods produced Gelman and Rubin statistics less than 1.2, which implies both methods reached convergence.

the nature of the challenge dataset. Although the challenge dataset has less pixels than the realistic or closely linear datasets, it will take BIMS and SBIMS longer to reach convergence for the challenge dataset because we are considering 8 possible chemicals in the challenge dataset, whereas we only considered 3 and 5 possible chemicals in closely linear and realistic datasets, respectively. We ran the MCMC algorithm under BIMS for 10,000 iterations with a burn-in of 5,000 for inference, which took 7.7 hours in total or 2.8 seconds per iteration. We ran two MCMC chains for BIMS, and we show log posterior distribution values for both chains in Figure 3.8. Using the first 5,000 iterations in each of the two chains, we calculated a Gelman and Rubin statistic of 1.0012, which implies convergence since it is less than 1.2. Since the SBIMS method is more complex than BIMS, SBIMS reached convergence in 180,000 iterations which took a total computational time of 6.25 days, or 3 seconds per iteration. We then ran SBIMS for an additional 10,000 iterations which were used for inference. We also ran two MCMC chains for SBIMS, and we show log posterior distribution values in the Figure 3.8. Using all 180,000 iterations, the two chains yielded a Gelman and Rubin statistic less than 1.2, specifically, 1.1162, which implies convergence was reached.

3.7 Discussion

Our illustrations in Chapters 2 and 3 clearly show the advantages of both using Bayesian model selection and of including spatial information for the detection and identification of chemical plumes in hyperspectral images. There are several important differences between the FL and SBIMS, including that FL is a penalization method, and SBIMS is a Bayesian method that uses priors and takes into account collinearity among chemicals. However, from a purely spatial modeling perspective, another important difference between the two methods is that SBIMS places spatial information indirectly through the prior inclusion probability of each variable rather than directly through the prior on the coefficients (as is implicitly done with the fussed LASSO). This spatial difference as well as other differences between the FL and SBIMS yields more accurate estimators under the SBIMS method. Furthermore, the use of a clipped Gaussian Markov random field to induce a prior on the inclusion probabilities proved to be flexible.

Among the three datasets considered in this chapter and in Chapter 2, we recognize that in general all methods performed the best on the closely linear dataset, performed less well on the realistic dataset, and yielded the poorest results on the challenge dataset. The differences in results among the three datasets can be attributed to several reasons, including the geography of the hyperspectral image (i.e., downward looking images vs side looking images), the size of the library of possible chemicals (i.e., three for closely linear dataset, five for realistic dataset, and eight for challenge dataset), and chosen embedding parameters for the different semi-synthetic datasets. In particular side viewing data cubes can be more difficult to analyze since the estimate of the atmosphere's transmittance is typically less reliable for side-viewing cubes than for downward looking cubes. This is because for downward looking images the atmosphere's transmittance can be estimated directly from the data cube (see Section 1.3), whereas this is not possible for side viewing cubes. In the case of side-viewing hyperspectral images, "cookie-cutter" estimates for the atmosphere's transmittance are typically used. The results of the realistic dataset in comparison with the closely linear dataset do show that considering a larger library of possible chemicals does affect detection and identification accuracy, especially for the LASSO, BIMS, and FL methods. Furthermore regarding embedding parameters, we mentioned previously that we hypothesize that the plumes in the Challenge dataset were embedded using a Gaussian mask which can make it more difficult to yield accurate detection and identification results than a uniform embedding mask. Note that we used uniform embedding masks for the plumes in the closely linear and realistic datasets. Additionally, it is also possible that the plumes for the challenge dataset were embedded at plume temperatures and with chemical concentration levels that were more challenging than values we used for the embedding parameters in the closely linear and realistic datasets.

One of the main shortcomings of our methodology is that it relies on a linear approximation to non-linear physical phenomena. Although the assumptions underlying this approximation are often valid in many practical circumstances, it is clear that future work will need to consider methods that can cover scenarios where a plume is not optically thin, where the background is not flat, or where the temperature difference between the plume and background is large. In particular, we examine methods that can address cases when the plume is not optically thin in the next chapter.

Chapter 4

Supervised non-linear model selection methods for chemical plume detection and identification in hypserspectral imagery

In this chapter we develop a novel Bayesian approach to detection and identification when linearizing assumptions (see Section 2.1) cannot be considered valid. Without linear assumptions a convenient linear unmixing model, as discussed in Section 2.2, is no longer applicable, and instead we must un-mix a nonlinear signal composed of light coming from the plume and the background. To perform nonlinear un-mixing we borrow ideas from the uncertainty quantification literature and propose modeling the signal as a non-parametric regression, where we place a prior on the regression function. In particular we propose placing a Gaussian Process (GP) prior on the regression function that is centered on the linear un-mixing model. With this approach we have two variance terms, one that explains linear variance in the signal, and one that explains nonlinear variance. Furthermore, variable selection is incorporated in the GP prior by carefully structuring the prior on the covariance function. Similar to the BIMS and SBIMS models (see sections 2.4 and 3.2), we place a mixture of g-priors on the coefficients of the linear model that serves to center our nonparametric formulation. Furthermore, we use the Jefferey's prior on the linear model variance, and employ Markov Random Fields to enforce spatial structure in the model. Therefore the models we evaluate in this section extend BIMS and SBIMS to be non-parametric, and in the sequel we will refer to them as non-parametric BIMS (NPBIMS), and spatial non-parametric BIMS (SNPBIMS).

Gaussian processes priors have a long history in many areas of statistics, especially in spatial statistics, where such methods are referred to as "kriging" (Matheron, 1963). Neal (1997) and Rasmussen (2004) are standard references for the use of GP priors in regression, however we can see GP priors placed on regression functions as early as O'Hagan and Kingman (1978). When using GP priors, a covariance function between data points must be specified. The use of different covariance functions, and the different ways to specify parameters within covariance functions, give GP priors the flexibility to fit a wide variety of curves, especially non-linear curves. Abrahamsen (1997) gives a thorough review of covariance functions in GP's and their properties. For the application of detection and identification, we use a covariance function that can be adapted for model selection, e.g. see Linkletter et al. (2006a); Sacks et al. (1989). Additionally this form of covariance function makes the NPBIMS and SNPBIMS computationally feasible via MCMC sampling.

4.1 Non-linear likelihood

Recall the equation of the at-sensor radiance model when a plume is present from Section 1.2,

$$L_{\text{on}}(\lambda) = L_a(\lambda) + \tau_a(\lambda)L_p(\lambda) + \tau_a(\lambda)\tau_p(\lambda)L_b(\lambda). \tag{4.1}$$

Non-linearities in the model exist in the light source models, L_a, L_p, and L_b, which are modeled using the Plank function (see equation (1.2)). The transmittance of the plume is also a non-linear function known as Beers law,

$$\tau_p(\lambda) = \exp\left(-\sum_{m=1}^{M} \gamma_m \alpha_m(\lambda)\right),$$

where $\alpha_m(\lambda)$ are high resolution chemical spectral signatures, and γ_m is the concentration of chemical m present in the plume. By adding and subtracting $\tau_a(\lambda)L_b(\lambda)$ to the right hand side of equation 4.1, we can rewrite the radiance model as,

$$L_{\text{on}}(\lambda) = \tau_a(\lambda)(L_p(\lambda) - \{1 - \tau_p(\lambda)\}L_b(\lambda)) + L_{\text{off}}(\lambda), \tag{4.2}$$

where $L_{\text{off}}(\lambda) = L_a(\lambda) + \tau_a(\lambda)L_b(\lambda)$, and is the at sensor-radiance model when no plume is present, i.e, the background (see section 1.2 for more details). Equation (4.2) is also a signal plus clutter model, where the first term on the right hand side is the signal, and as before, the light from the background is the clutter. Unlike in Section 2.1, the signal no longer takes on a convenient linear form. To address this we model the pixel spectrum as a general function. As before, the hyperspectral sensor captures the at-sensor radiance signal as shown in equation (4.1) on a series of I adjacent spectral bands centered at wavelengths, $\lambda_1, ..., \lambda_I$, and then processes it through its spectral response function, $R_F(\lambda)$ and introduces noise, $n(\lambda)$ to produce the pixel spectrums in a hyperspectral image, $\boldsymbol{x} = (x_1, ..., x_I)$, which we now model as

$$x_i = f(s_{1,i}, ..., s_{M,i}) + v_i, \quad i = 1, ..., I. \tag{4.3}$$

In other words, the pixel spectrum is modeled as some function of the down sampled chemical spectral signatures, $s_{m,i} = \alpha_m(\lambda_i) * R_F(\lambda_i)$, plus the background and sensor noise, $v_i = L_{\text{off}}(\lambda_i) * R_F(\lambda_i) + n(\lambda_i)$. Let $f_i = f(s_{1,i}, ..., s_{M,i})$, and note that in a hyperspectral image we have J pixels. Then we can rewrite model in (4.3) in vector form for one pixel as,

$$\boldsymbol{x}_j = \boldsymbol{f}_j + \boldsymbol{v}_j, \qquad \boldsymbol{v}_j \sim N(\boldsymbol{\mu}, \sigma_j^2 \boldsymbol{\Sigma}), \tag{4.4}$$

where we assume the error term is normally distributed. As before, $\boldsymbol{\mu}$ is the background mean, and $\boldsymbol{\Sigma}$ is the background covariance. To ease computation, we proceed by using point estimates for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Sigma}}$. As before, $\hat{\boldsymbol{\mu}}$ is the mean signal over all pixels, while $\hat{\boldsymbol{\Sigma}}$ is estimated from the whole image using dominant mode rejection (Manolakis et al., 2009).

4.2 Nonparametric BIMS

In this section we discuss the NPBIMS method. To build NPBIMS first recall from Section 2.2 that the mean of the linear model minus the background mean is $\boldsymbol{S}\boldsymbol{g}_j$, which can be understood as the linear plume signal. The matrix, $\boldsymbol{S}$, has dimension $I \times M$ and its columns are the unwhitened downsampled chemical spectral signatures, and $\boldsymbol{g}_j = (g_{1,j}, ..., g_{M,j})$ are the corresponding regression coefficients for pixel j. Also recall that for the BIMS method we introduced latent variabes, $\boldsymbol{\omega}_j = (\omega_{1,j}, ..., \omega_{M,j})$ for each pixel $j = 1, ...J$, where $\omega_{m,j} = 0$ if $g_{m,j} = 0$, and $\omega_{m,j} = 1$, otherwise. We proceed by placing a GP prior on the function $\boldsymbol{f}_j$ in the likelihood in equation (4.4) that is centered on the linear plume signal,

$$\boldsymbol{f}_j | \boldsymbol{\omega}_j \sim \mathrm{GP}(\boldsymbol{S}(\omega_j)\boldsymbol{g}_j(\omega_j), \kappa_j \sigma_j^2 C(\boldsymbol{\psi}_j, \boldsymbol{\omega}_j)). \tag{4.5}$$

For a finite sample the prior in (4.5) implies the joint normal prior,

$$\boldsymbol{f}_j = \boldsymbol{S}(\boldsymbol{\omega}_j)\boldsymbol{g}_j(\boldsymbol{\omega}_j) + \boldsymbol{\nu}_j, \qquad \boldsymbol{\nu}_j \sim N(0, \kappa_j \sigma_j^2 C(\boldsymbol{\psi}_j, \boldsymbol{\omega}_j)). \tag{4.6}$$

The matrix, $\boldsymbol{S}(\boldsymbol{\omega}_j)$ includes only the columns of $\boldsymbol{S}$ where $\omega_{m,j} = 1$, and $\boldsymbol{g}_j(\boldsymbol{\omega}_j)$ is the subvector of $\boldsymbol{g}_j$ corresponding to the non-zero values in $\boldsymbol{\omega}_j$. The parameter, κ_j can be interpreted as the ratio between the observational noise and the variance of the GP prior around its mean. Note that when $\kappa_j = 0$ for all j we return to the BIMS model.

In our model, the correlation function, $C(\boldsymbol{\psi}_j, \boldsymbol{\omega}_j)$ takes the form,

$$C_{h,i}(\boldsymbol{\psi}_j, \boldsymbol{\omega}_j) = \prod_{\omega_{j,m}=1} \psi_{m,j}^{2^{\beta_{m,j}} |s_{h,m} - s_{i,m}|^{\beta_{m,j}}}, \tag{4.7}$$

where $\psi_{m,j}$ can take on values in $(0, 1]$. Values close to 1 imply that the process in pixel j does not depend on chemical, m. For this reason, we use a spike-and-slab prior (George and

McCulloch, 1993) on $\boldsymbol{\psi}_j$,

$$p(\boldsymbol{\psi}_j|\boldsymbol{\omega}_j) = \prod_{m=1}^{M} \omega_{m,j}\mathbf{1}(0 < \psi_{m,j} < 1) + (1-\omega_{m,j})\delta_1(\psi_{m,j}), \tag{4.8}$$

where δ_1 is the Dirac delta function at 1. The parameter, $\beta_{m,j}$ controls the smoothness of the GP prior. For example, $\beta_{m,j} = 2$ leads to realizations that are mean-square infinitely differentiable. Because this parameter is difficult to estimate, we set $\beta_{m,j} = 1.9$ for all $m = 1, ..M$ and all $j = 1, ..., J$ in to avoid computational issues that can arise from setting $\beta_{m,j} = 2$.

To gain more intuition into the covariance function, $C_{h,i}$, note that it will produce an $I \times I$ covariance function. For simplicity, imagine we are only considering one chemical, i.e., $M = 1$, and assume that its corresponding indicator variable, $\omega_{m,j}$, is equal to 1. We can now drop the product operator in (4.7). Therefore, the resulting $I \times I$ covariance matrix will express the covariance between each element in the one chemical signature, $\boldsymbol{s}_m$, that we are considering. If $\psi_{m,j} = 1$, then we will obtain a rank-1 covariance matrix. For more details please refer to Linkletter et al. (2006b).

We continue by placing mixtures of g-priors on $\boldsymbol{g}_j$ to address colinearity among chemical signatures,

$$\boldsymbol{g}_j|c_j, \sigma_j^2, \boldsymbol{\omega}_j \sim N\left(\boldsymbol{g}_j(\boldsymbol{\omega}_j)\,|\,0, c_j\sigma_j^2\boldsymbol{S}'(\boldsymbol{\omega}_j)\hat{\Sigma}^{-1}\boldsymbol{S}(\boldsymbol{\omega}_j)\right)\prod_{m:\omega_{m,j}=0}\delta_0(g_{m,j}), \tag{4.9}$$

and we place the Jeffrey's prior on $\sigma_j^2, p(\sigma_j^2) \propto \frac{1}{\sigma_j^2}$. These choices of priors on $\boldsymbol{f}_j, \boldsymbol{g}_j$, and σ_j^2 allows us to integrate out these parameters resulting in a likelihood of the form,

$$p(\boldsymbol{x}_1, ..., \boldsymbol{x}_J|\kappa_1, ..., \kappa_J, c_1, ..., c_J, \boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_J, \boldsymbol{\psi}_1, ..., \boldsymbol{\psi}_J) \propto$$
$$\prod_{j=1}^{J}\boldsymbol{L}^{-1/2}(\kappa_j, \boldsymbol{\psi}_j, c_j, \boldsymbol{\omega}_j)\left((\boldsymbol{x}_j - \hat{\mu})'\boldsymbol{L}^{-1}(\kappa_j, \boldsymbol{\psi}_j, c_j, \boldsymbol{\omega}_j)(\boldsymbol{x}_j - \hat{\mu})\right)^{-\frac{I}{2}}, \tag{4.10}$$

where $\boldsymbol{L}(\kappa_j, \boldsymbol{\psi}_j, c_j, \boldsymbol{\omega}_j) = \hat{\Sigma} + \kappa_j C(\boldsymbol{\psi}_j, \boldsymbol{\omega}_j) + c_j\boldsymbol{S}(\boldsymbol{\omega}_j)\left(\boldsymbol{S}'(\boldsymbol{\omega}_j)\hat{\Sigma}^{-1}\boldsymbol{S}(\boldsymbol{\omega}_j)\right)^{-1}\boldsymbol{S}'(\boldsymbol{\omega}_j)$.

As with BIMS, we again assume that chemical inclusion probabilities on a given pixel are independent and uniform, which is equivalent to assuming independent Beta-Binomial priors on $\boldsymbol{\omega}_j$ as shown in (2.8). Also similar to BIMS, we place an inverse gamma prior on c_j, $p(c_j) =$

$\mathsf{IGam}(.5, .5I)$. Lastly, we place an inverse gamma prior on κ_j, $p(\kappa_j) = \mathsf{IGam}(a_\kappa, b_\kappa)$, where extensive prior sensitivity analysis was performed for prior choices for a_κ, and b_κ. In general the prior choices of a_κ and b_κ depends on the hyperspectral image being analyzed. We explored making b_κ random while fixing $a_\kappa = 11$, however we encountered identifiability issues with b_κ under such a setup. We discuss the prior choices of a_κ and b_κ further in results sections 4.4, and 4.5.

4.3 Spatial NPBIMS

Like the LASSO and BIMS methods (see Chapter 2), an obvious shortcoming of NPBIMS is that pixels are treated as independent and spatial information in the image is ignored. In this section we describe the Spatial NPBIMS (SNPBIMS) model, which extends NPBIMS to incorporate spatial dependence among pixels.

We proceed in the same manner as in Section 3.2 where we described the SBIMS method that extended the BIMS method to incorporate spatial information. As before, we consider a first order neighborhood structure in the neighborhood matrix, $\boldsymbol{\Upsilon}$, where $\upsilon_{j,j'} = 1$ if an only if pixels j and j' are neighbors, and $\upsilon_{j,j'} = 0$, otherwise. We then introduce latent variables, $\boldsymbol{\ell}_m = (\ell_{m,1}, ..., \ell_{m,J})$, on which we place a clipped Gaussian Markov random field prior,

$$\ell_{m,j} \mid \ell_{m,1}, \cdots, \ell_{m,j-1}, \ell_{m,j+1}, \cdots, \ell_{m,J}, \alpha_m, \rho_m, u_m \sim \mathsf{N}\left(\{1-\rho_m\}\alpha_m + \rho_m \frac{\sum_{j'\neq j} \upsilon_{j,j'}\ell_{m,j}}{\sum_{j'\neq j}\upsilon_{j,j'}}, \frac{1}{u_m \sum_{j'\neq j}\upsilon_{j,j'}}\right), \quad (4.11)$$

then let $\omega_{m,j}|\ell_{m,j} = 1$ if $\ell_{m,j} > 0$, and $\omega_{m,j}|\ell_{m,j} = 0$, otherwise. Again, this prior incorporates spatial information by smoothing neighboring $\ell_{m,j}$'s to be similar. The amount of spatial smoothing is controlled by ρ_m and u_m, where a value of $\rho_m = 0$ implies no spatial dependence, and we are are returned to a model similar to NPBIMS. We set $0 < \rho_m < 1$ to ensure the joint prior,

$$p(\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_M \mid a_m, u_m, \rho_m) = \prod_{m=1}^{M} \mathsf{N}\left(\alpha_m \mathbf{1}, \{u_m(\boldsymbol{D}_\Upsilon - \rho_m \boldsymbol{\Upsilon})\}^{-1}\right),$$

implied by equation (4.11), is proper. This also guarantees that the posterior distribution of the model is proper. The matrix, $\boldsymbol{D}_\Omega$, is defined as $\boldsymbol{D}_\Upsilon = \text{diag}\left\{\sum_{j\neq 1}\upsilon_{j,1}, \ldots, \sum_{j\neq J}\upsilon_{j,J}\right\}$.

Recall that under this setup the marginal prior probability that chemical m exists in pixel j is,

$$\theta_{m,j} = \Phi\left(\frac{u_m^{1/2}\alpha_m}{q_{m,j,j}^{1/2}}\right),$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function, and $q_{m,j,j}$ is the j-th diagonal term of $\boldsymbol{Q}_m = \{\boldsymbol{D}_\Upsilon - \rho_m \boldsymbol{\Upsilon}\}$. To account for multiplicities we treat this probability as random by assigning priors on α_m, ρ_m, and u_m. In particular we assign,

$$\alpha_m \sim \mathsf{N}(0,1), \qquad u_m \sim \mathsf{Gam}(2,8), \qquad \rho_m \sim \mathsf{Uni}\{0.5, 0.75, 0.9, 0.95, 0.975, 0.99, 0.999\}.$$

These choices ensure an approximately uniform prior on the chemical inclusion probabilities, $\theta_{m,j}$. For more details see Section 3.2.

It is important to note that NPBIMS and SNPBIMS are more computationally demanding than their parametric counterparts, BIMS and SBIMS. We discuss this and other computational considerations more in Chapter 5.

4.4 Performance evaluation on nonlinear dataset

To evaluate NPBIMS and SNPBIMS we will first implement them on the nonlinear dataset described in Section 1.5. We focus on the nonlinear dataset as opposed to the closely linear and realistic datasets since we embedded plumes in the nonlinear dataset such that linear assumptions do not hold, specifically linearization of Beer's Law (see Section 1.4.3). As will be explained in more detail later on in this section, our nonparametric identification methods are very computationally burdensome and require long run times to reach convergence. For the nonlinear dataset (which has 89,600 pixels) we estimate it would take months for SNPBIMS to reach convergence. In Chapter 6, we discuss that developing ways to improve NPBIMS and SNPBIMS' computationally efficiency is part of future work. Hence, we will evaluate the nonparametric methods on the nonlinear dataset in a two stage process. First we will implement BIMS and SBIMS on the whole nonlinear dataset. This will act as a first-pass detection that will detect no-plume regions and target plume regions. We will then implement NPBIMS and SNPBIMS on only the target plume regions.

4.4.1 First pass detection via BIMS and SBIMS on nonlinear dataset

Figure 4.1 shows the identification results under BIMS in the left panel, and the identification results under SBIMS in the right panel. We also show the results in confusion matrix form in Figure 4.2. Note that we ran BIMS for 10,000 MCMC iterations and SBIMS for 150,000 MCMC iterations, which was the number of iterations that reached convergence for the closely linear dataset. BIMS took 2.78 hours to complete, or 1 second per iteration, and SBIMS took 3 days to complete, or 1.7 seconds per iteration. We used the last 5,000 iterations of the BIMS MCMC algorithm for inference. Whereas for SBIMS we ran an additional 10,000 iterations, which were used for inference. We show log posterior values over the iterations under both BIMS and SBIMS in Figure 4.3, and in particular we ran two chains for SBIMS.

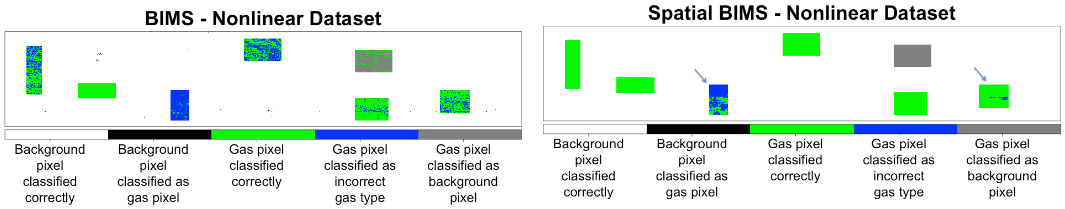
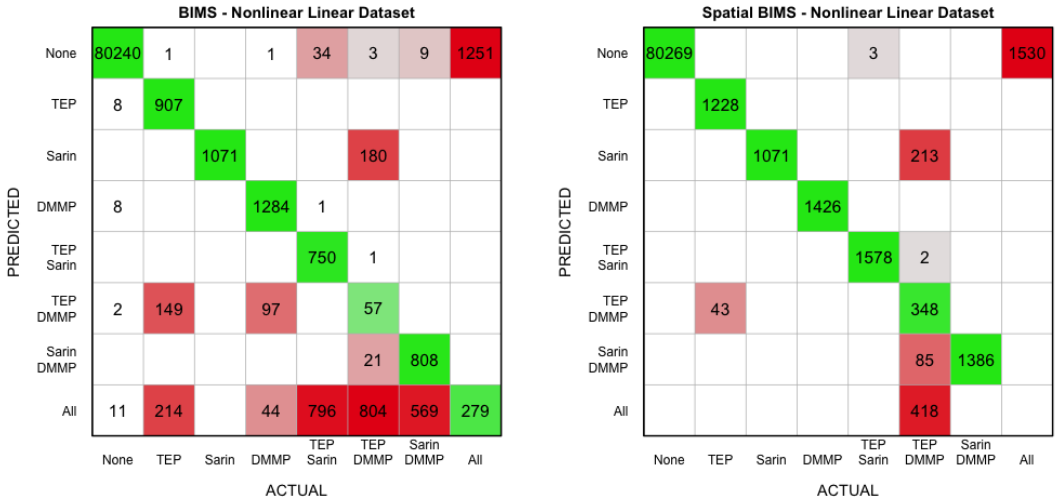


Figure 4.1: Identification results for BIMS and SBIMS on nonlinear dataset mapped onto the area covered by the synthesized data. Two plumes that have room for improvement are pointed out in the right plot.


BIMS - Nonlinear Linear Dataset

PREDICTED \ ACTUAL	None	TEP	Sarin	DMMP	TEP Sarin	TEP DMMP	Sarin DMMP	All
None	80240	1		1	34	3	9	1251
TEP	8	907						
Sarin			1071			180		
DMMP	8			1284	1			
TEP Sarin					750	1		
TEP DMMP	2	149		97		57		
Sarin DMMP						21	808	
All	11	214		44	796	804	569	279

Spatial BIMS - Nonlinear Linear Dataset

PREDICTED \ ACTUAL	None	TEP	Sarin	DMMP	TEP Sarin	TEP DMMP	Sarin DMMP	All
None	80269				3			1530
TEP		1228						
Sarin			1071			213		
DMMP				1426				
TEP Sarin					1578	2		
TEP DMMP		43				348		
Sarin DMMP						85	1386	
All						418		

Figure 4.2: Confusion matrices for BIMS and NPBIMS on the nonlinear dataset.

We would like to highlight the fact that BIMS and SBIMS perform worse on the nonlinear dataset than on the closely linear dataset (see the bottom panel of Figure 2.3 and bottom panel

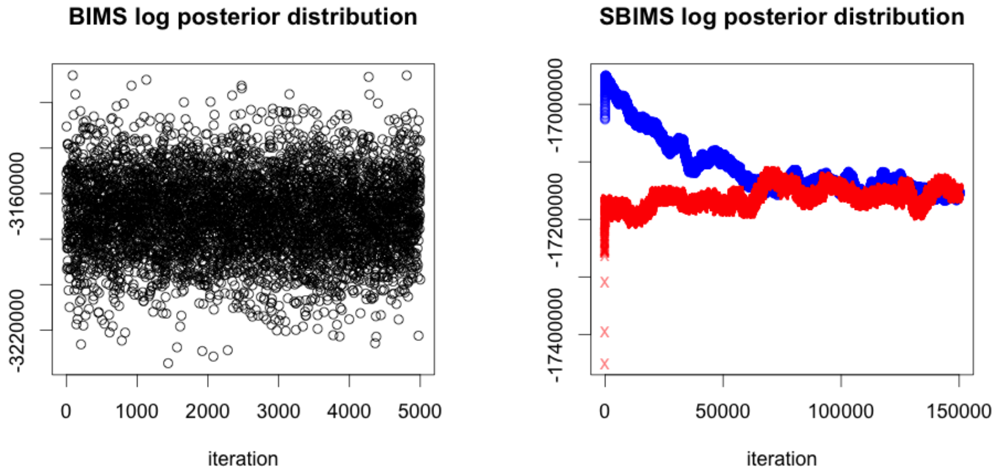


Figure 4.3: Trace plots of log posterior distribution under BIMS (left plot) and SBIMS (right plot) for nonlinear dataset.

of Figure 3.3). BIMS attained ACC and MCC values of 0.9972 and 0.9859, respectively on the closely linear dataset, whereas for the nonlinear dataset BIMS attained ACC and MCC values of 0.9531 and 0.7471, respectively. Furthermore, SBIMS preformed practically perfectly for the closely linear dataset where ACC and MCC values of 0.9998 and 0.9999 were attained, but under the nonlinear dataset SBIMS only attained ACC and MCC values of 0.9744 and 0.8627, respectively. This difference in performance is clearly related to the use of a non-linear regime for the embedding process (see discussion in Section 1.4).

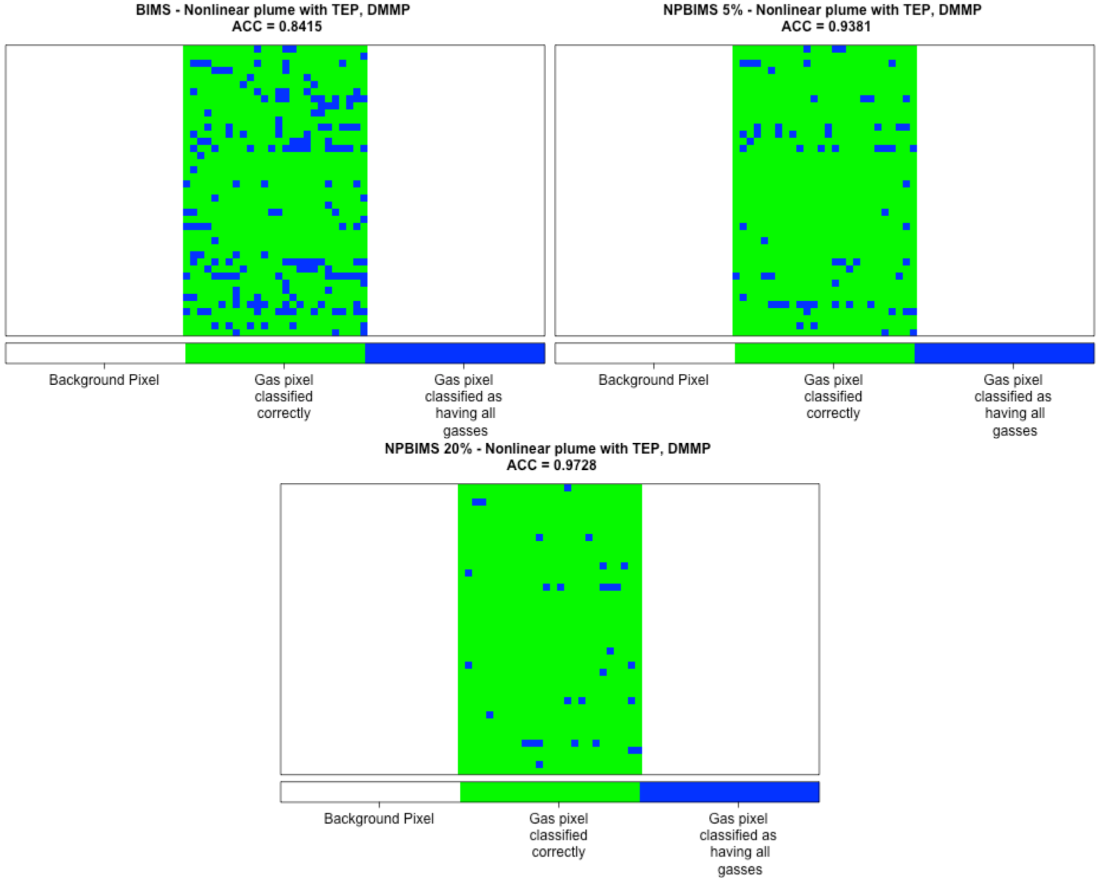
As expected, we see that SBIMS performs better than BIMS on the nonlinear dataset. In fact SBIMS is able to identify some plumes perfectly despite nonlinearity in the plumes. On the other hand, we point out two plumes in the right panel of Figure 4.1 that have room for improvement. The first pointed out plume on the left contains both TEP and DMMP, and recall that these two chemicals have spectral signatures with high correlation (see section Figure 1.7, and Table 1.1 in Section 1.5). The second pointed out plume on the right contains TEP only. As the second stage of our process, we analyze these two plumes under NPBIMS and SNPBIMS to evaluate the effectiveness of the nonparametric identification methods. Note that although the gray region in the right panel of Figure 4.1 is an embedded plume, NPBIMS identified most of these pixels as having no plume, which might explain why SBIMS smoothed the entire plume region as having no gas. Therefore, we cannot realistically consider it as a plume region needing improvement.

4.4.2 NPBIMS on the Nonlinear TEP and DMMP Plume

We first start with the plume that had room for the most improvement. Since we performed a first pass detection in the previous section, we can already improve results in two ways. First, we can re-estimate $\mathbf{\Sigma}$ and $\boldsymbol{\mu}$. Recall that $\mathbf{\Sigma}$ and $\boldsymbol{\mu}$ are the background covariance and background mean, and prior to any detection or identification, their estimates, $\hat{\mathbf{\Sigma}}$ and $\hat{\boldsymbol{\mu}}$, are calculated using all pixels (see Section 2.2) and are therefore contaminated by plume present pixels. After the first detection pass with BIMS and SBIMS, we can now get less contaminated estimates by using only the no-plume detected pixels to calculate $\hat{\mathbf{\Sigma}}$ and $\hat{\boldsymbol{\mu}}$. Secondly, since we are analyzing a region that we know has plume, we can give the no gas model zero probability in the prior. With these improvements we will analyze the TEP/DMMP plume region from the nonlinear dataset under the nonparametric methods, NPBIMS and SNPBIMS, as well as the parametric methods, BIMS and SBIMS, for comparison.

In Figure 4.4 we show identification results for BIMS and two versions of NPBIMS. NPBIMS 5% assumes in the prior that nonlinearities can explain 5% of the variance in the pixel signal. Specifically for NPBIMS 5% we set $b_\kappa = 0.5$ and $a_\kappa = 11$ for the Inverse Gamma prior on κ_j (see Section 4.2). NPBIMS 20% on the other hand assumes in the prior that 20% of the variance in the pixel signal can be explained by non-linearities, and specifically we assign $b_\kappa = 2.0$ and $a_\kappa = 11$. We can see significant improvement of NPBIMS 5% over BIMS and slight improvement of NPBIMS 20% over NPBIMS 5%. The improvement of NPBIMS over BIMS is also evidenced by the ACC confusion matrix metrics provided in the plot titles. In particular BIMS was able to correctly identify about 84% of pixels correctly, whereas NPBIMS 5% and NPBIMS 20% were able to correctly identify 94% of pixels, and 97% of pixels, respectively. Note that each of the models identified pixels as either having TEP and DMMP (green pixels) or having all chemicals (blue pixels). The white pixels are background pixels that are not part of the target plume region, as explained previously, but we include them in the plots for better visual appearance of the figure. Also note that since there are only two types of chemical combinations that were identified, we do not show confusion matrices and feel the plots and ACC metrics in Figure 4.4 are sufficient.

For the results in Figure 4.4 we ran each model for 10,000 MCMC iterations with a burn-in of


BIMS - Nonlinear plume with TEP, DMMP
ACC = 0.8415

NPBIMS 5% - Nonlinear plume with TEP, DMMP
ACC = 0.9381

NPBIMS 20% - Nonlinear plume with TEP, DMMP
ACC = 0.9728

Figure 4.4: Identification results of BIMS and two versions of NPBIMS on the TEP/DMMP target plume region from the nonlinear dataset.

5,00 iterations for inference. Keep in mind that this particular target region has only 1,066 total pixels, whereas the full nonlinear dataset has 89,600 pixels in total. We show the log posterior values under each model for the last 5,000 iterations in Figure 4.5. For BIMS it took a total of 16.67 minutes, or 0.1 seconds per iteration to complete. Both NPBIMS 5% and NPBIMS 20% took 2.5 hours in total, or .9 seconds per iteration to complete. Notice that NPBIMS is much more computationally burdensome than BIMS mainly because there are large matrix multiplication operations necessary for the NPBIMS model, but are not required under BIMS. Therefore, as mentioned previously, NPBIMS has the disadvantage of needing long run times.

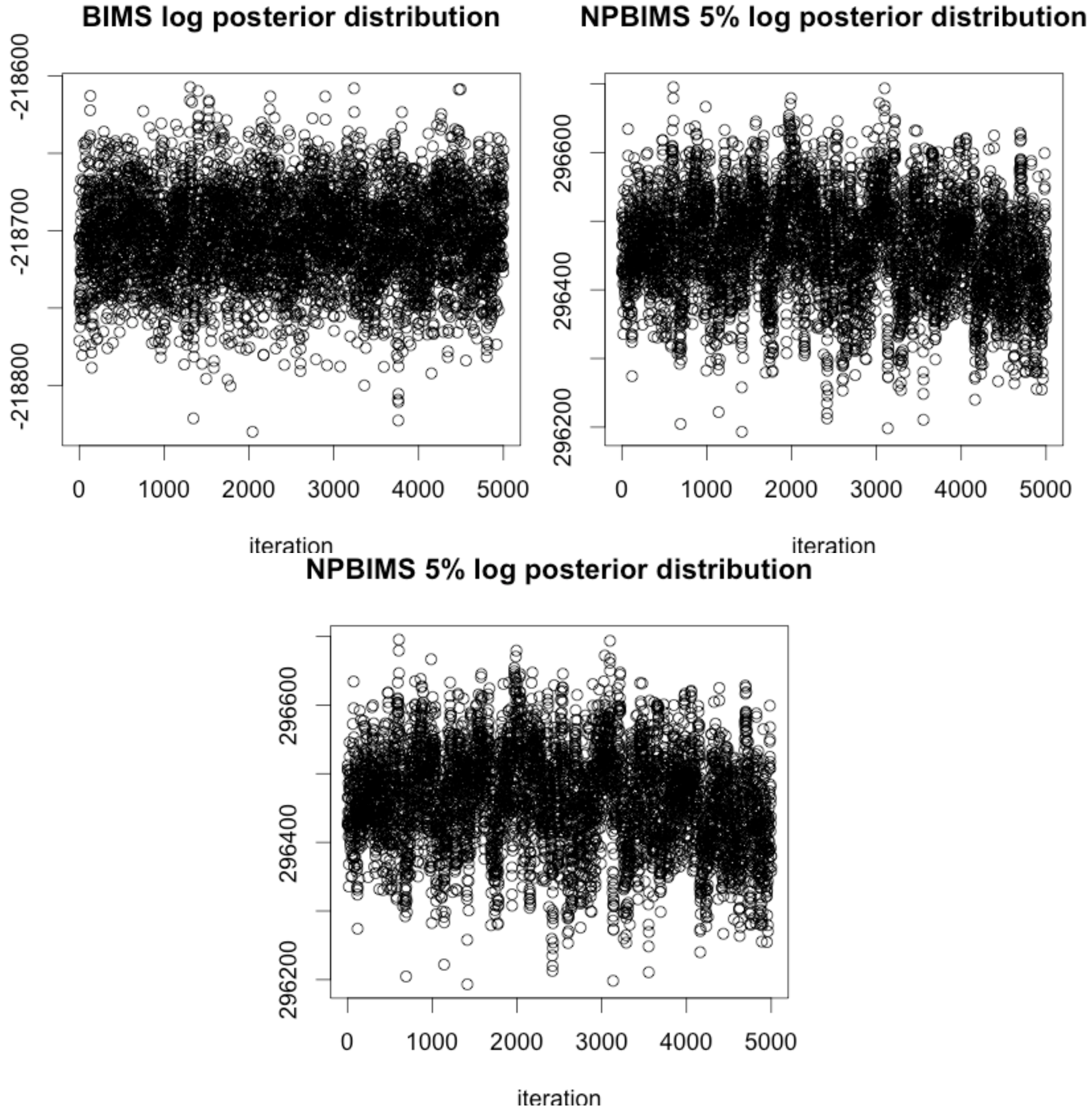


Figure 4.5: Trace plots of BIMS and two version of NPBIMS for the TEP and DMMP target plume from the challenge dataset.

4.4.3 SNPBIMS on the Nonlinear TEP and DMMP Plume

In Figure 4.6 we show identification results for SBIMS, and two versions of SNPBIMS on the same TEP/DMMP plume region from the nonlinear dataset. We can see that SBIMS performs perfectly, i.e., it is able to correctly identify 100% of pixels in the target area. On the hand, the SNPBIMS models are making some mistakes, which are all located at the bottom boundary of the plume. SNPBIMS 5% is the spatial extension of NPBIMS 5% (see section 4.3), and we can see it performs slightly worse than SBIMS as it able to correctly identify only 98.12%

of pixels correctly. Similarly, SNPBIMS 20%, which is the spatial extension of NPBIMS 20%, also performs slightly worse than SBIMS by correctly identifying only 98.22% of pixels. Note that both SNPBIMS model versions identify pixels as having either the correct combination of chemicals (TEP and DMMP) as represented by green pixels, or having only TEP as represented by blue pixels. Recall from Figure 4.4 that NPBIMS 5% and NPBIMS 20% identified pixels as either TEP and DMMP present, or as having all chemicals present. So under SNPBIMS the incorrect chemical, Sarin, is being smoothed out. However, for some pixels located at the boundary of the plume, DMMP, which is actually present, is also being smoothed out.

For the SBIMS results in Figure 4.6 we ran the MCMC for 20,000 iterations with a burn in of 10,000 for inference. We show log posterior distribution values in the top left panel of Figure 4.7. SBIMS took 1 hour to complete, or 0.36 seconds per iteration. We ran both SNPBIMS models for 50,000 MCMC iterations with a burn in of 40,000 for inference. Log posterior distribution

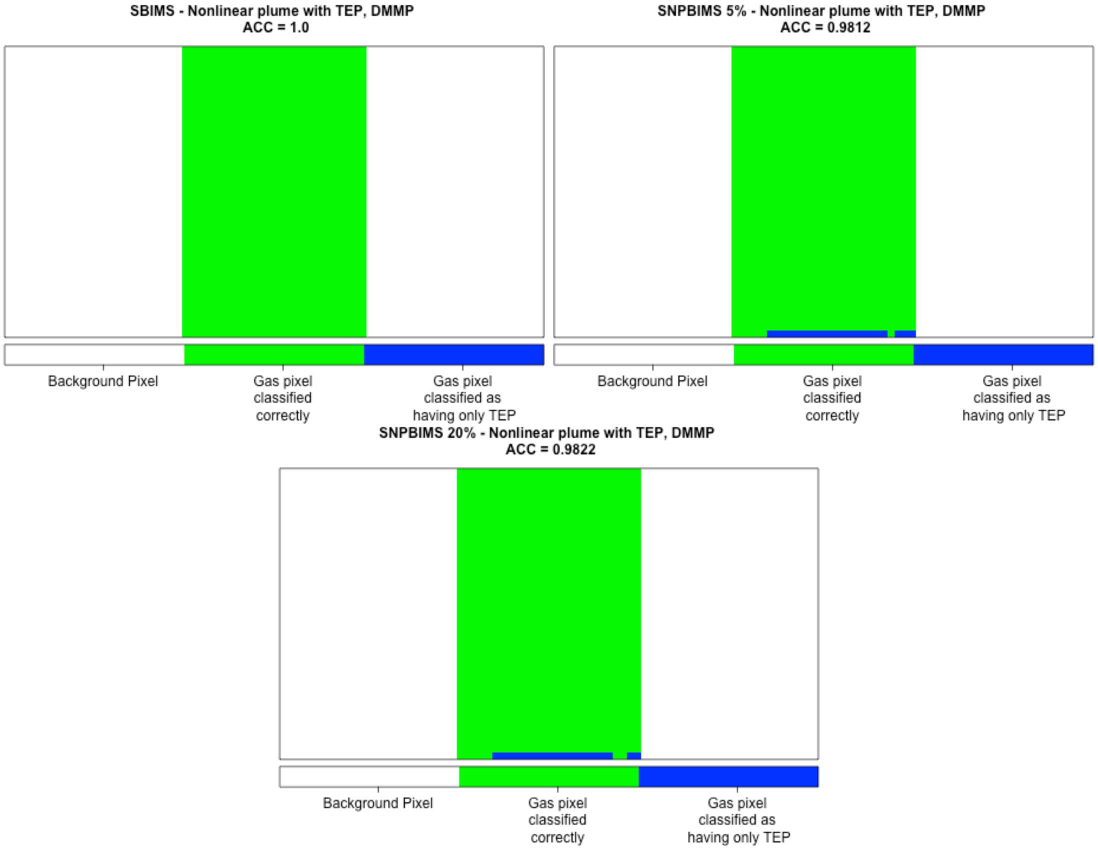
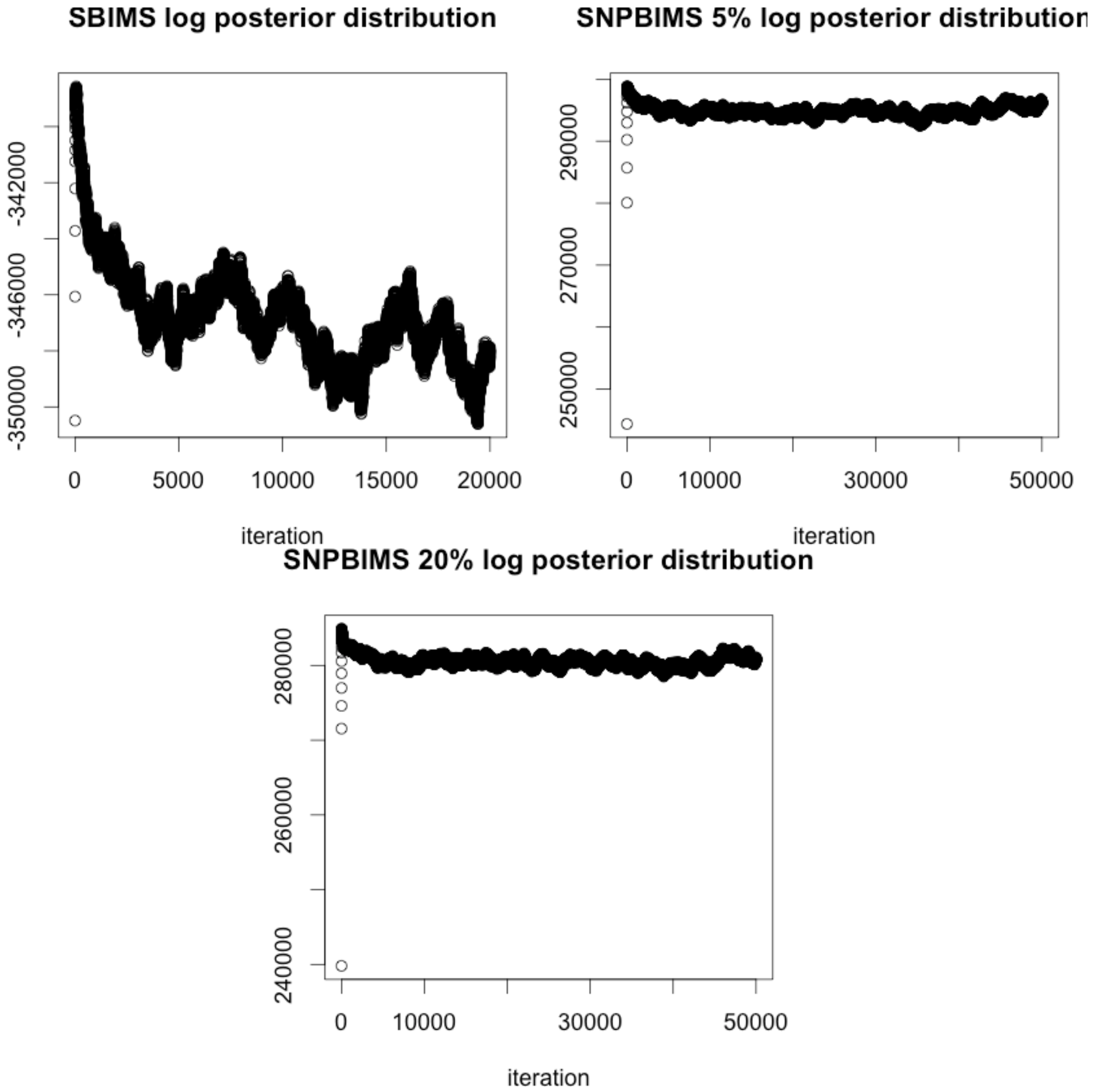


Figure 4.6: Identification results of BIMS and two versions of NPBIMS on the TEP/DMMP target plume region from the nonlinear dataset.



SBIMS log posterior distribution

SNPBIMS 5% log posterior distribution

SNPBIMS 20% log posterior distribution

Figure 4.7: Trace plots for SBIMS and two versions of SNPBIMS on the TEP/DMMP target plume region from the nonlinear dataset.

values under the SNPBIMS models are also shown in Figure 4.7. Both SNPBIMS 5% and SNPBIMS 20% took 13 hours to complete, or .99 seconds per iteration. Note how much longer the nonparametric method, SNPBIMS, takes when compared to its parametric counterpart, SBIMS, in terms of both total time and time per iteration. Similar to NPBIMS, SNPBIMS needs long run times because it requires large matrix-matrix operations that are not needed under SBIMS.

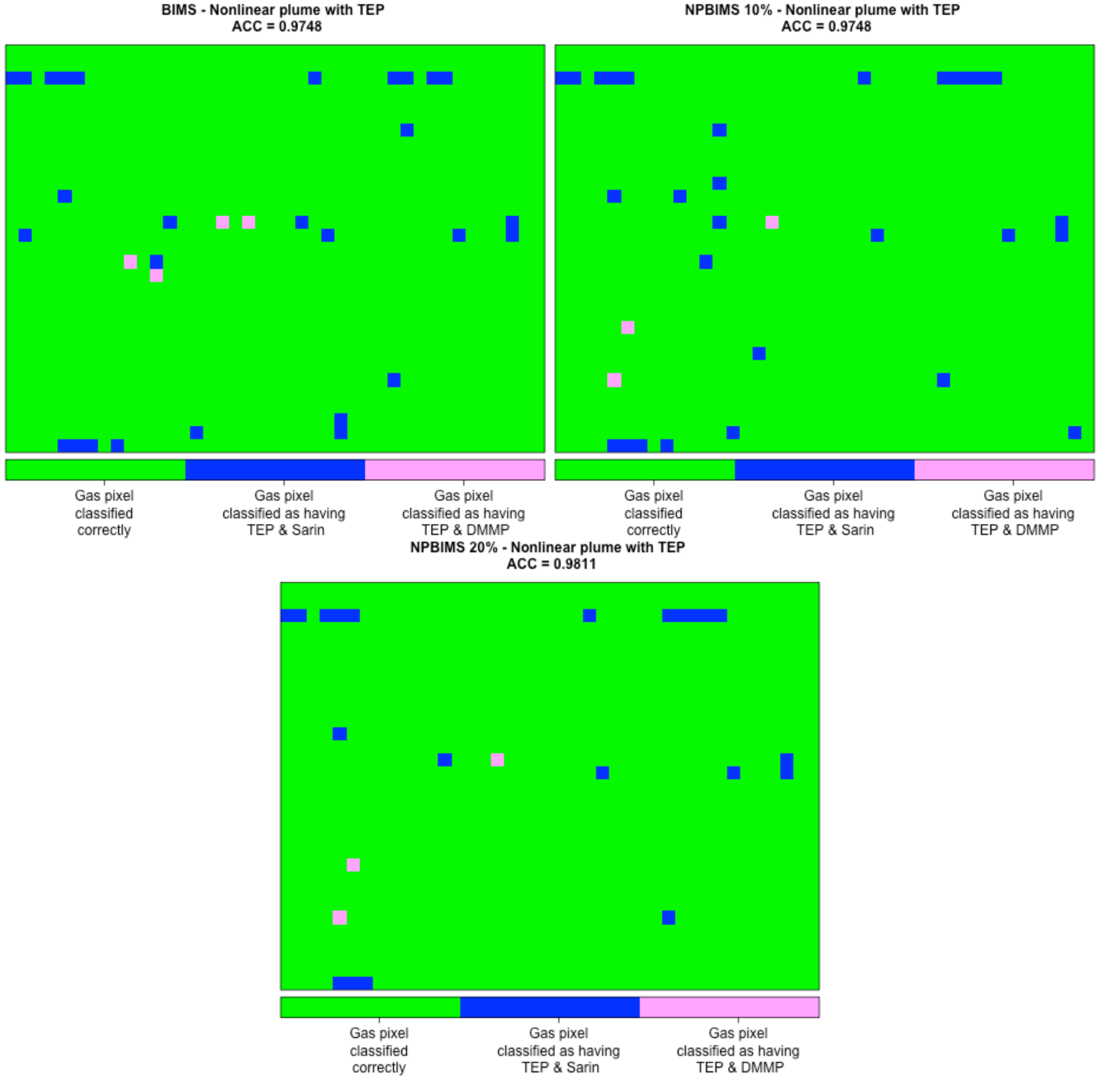


Figure 4.8: Identification results of BIMS, NPBIMS 10% and NPBIMS 20% on the TEP target plume region from the nonlinear dataset.

4.4.4 NPBIMS on the Nonlinear TEP Plume

In this section we evaluate the TEP plume target area from the nonlinear dataset using NPBIMS. As explained in Section 4.4.2, we again implement two improvements. First we use less contaminated estimates of $\boldsymbol{\Sigma}$ and $\boldsymbol{\mu}$, and we give the chemical model with no gasses zero probability in the prior. In Figure 4.8 we show the identification results of BIMS, NPBIMS 10%, and NPBIMS 20% for the TEP target plume. NPBIMS 10% assumes in the prior that nonlinearities can explain 10% of the variance in the pixel signal, and specifically we set $b_\kappa = 1.0$

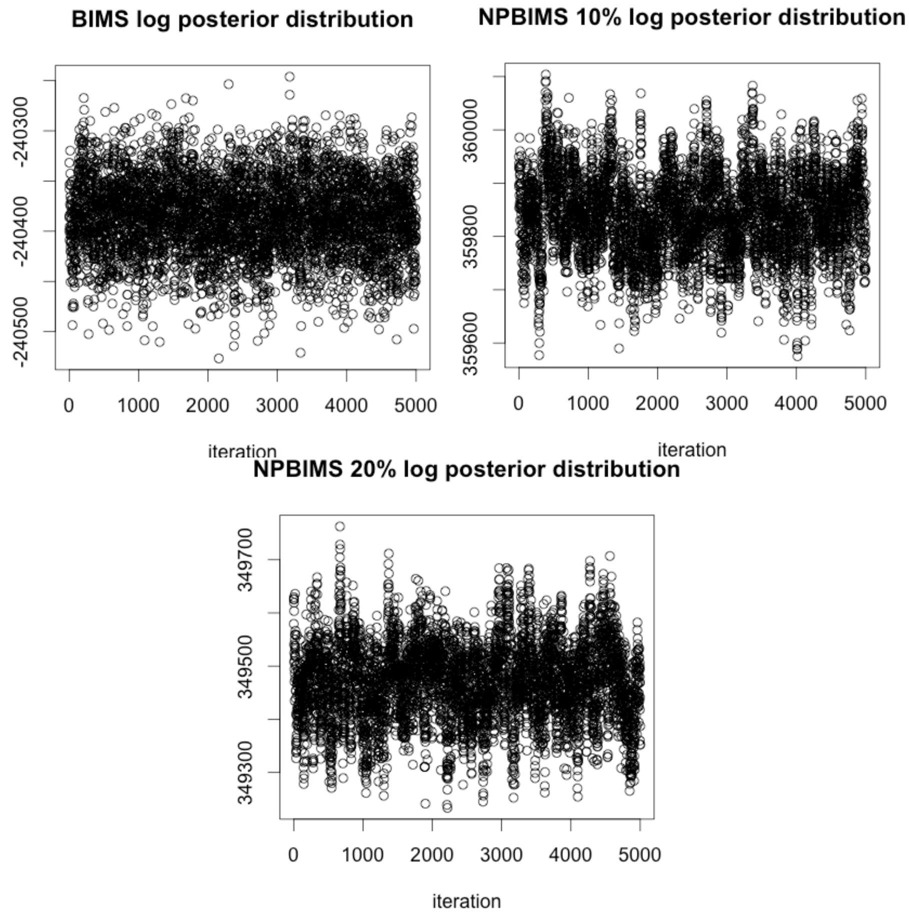


Figure 4.9: Trace plots for BIMS, NPBIMS 10%, and NPBIMS 20% on the TEP target plume region from the nonlinear dataset.

and $a_\kappa = 11$ for IG(a_κ, b_κ) prior on κ_j. And as in the previous section, NPBIMS 20% assumes in the prior that 20% of the pixel signal variance can be explained by nonlinearities, and we assign $b_\kappa = 2.0$ and $a_\kappa = 11$. BIMS attains a very high ACC value, i.e., it can identify 97.48% of pixels correctly. NPBIMS 10% has a similar performance to BIMS, and yields the same ACC value of BIMS. NPBIMS 20% performs somewhat better than BIMS as it identifies 98.11% of pixels correctly. Note that all 3 models identified pixels as either having TEP only (green pixels), which is the correct model for all pixels, having TEP and Sarin (blue pixels), or having TEP and DMMP (purple pixels).

For the TEP plume region we ran BIMS, NPBIMS 10%, and NPBIMS 20% each for 10,000 MCMC iterations with a burn in of 5,000 for inference. We show log posterior values over the iterations in Figure 4.9 for the last 5,000 iterations under each model. Similar to the BIMS implementation on the TEP/DMMP target plume, BIMS on the TEP plume took about 16 minutes to complete or .1 seconds. Note that the TEP plume has a total of 1,271 pixels, whereas the TEP and DMMP Plume had a total have 1,066 pixels. NPBIMS 10% and NPBIMS 20% on the other hand both took about 2.7 hours to complete or 1 second per iteration to complete.

4.4.5 SNPBIMS on the Nonlinear TEP Plume

In Figure 4.10 we show the identification results of SBIMS, SNPBIMS 10%, and SNPBIMS 20% on the TEP target plume, and we can see that all methods perform perfectly, meaning all methods are able to identify 100% of pixels correctly. Recall from Figure 4.8 that incorrectly classified pixels were identified as having TEP plus an extraneous chemical. Now both spatial parametric and spatial nonparametric methods are able to correctly smooth out the extraneous chemicals. Unlike the results of SNPBIMS on the TEP and DMMP plume in Figure 4.10, we do not see issues of over smoothing for the TEP plume. One reason for this is because the no-gas model has zero probability in the prior.

SBIMS was run for 20,000 with a burn in of 10,000 for inference. We show the log posterior distribution values under SBIMS in the top left panel of Figure 4.11. Similar to the implementation of SBIMS on the TEP and DMMP target plume, SBIMS took 1 hour to complete, or .36 seconds per iteration. We ran the SNPBIMS models each for 50,000 iterations with a burn in of 40,000 for inference. We also show log posterior distribution values for the two SNPBIMS models in the top right and bottom panels of Figure 4.11. Both versions of SNPBIMS took about 14 hours to complete, or about 1 second per iteration.

4.5 Performance evaluation on challenge dataset plumes

In this section we will evaluate NPBIMS and SNPBIMS on the challenge dataset. Like the previous section we also perform a two-stage process to avoid long run times. However, unlike the non-linear dataset, we now assume that a perfect first-pass detection was performed, where plumes 1, 2 and 3 in Figure 1.10 are detected as target plume regions, and the rest of the pixels are detected as having no plume. We will then implement the nonparametric methods on the plume target regions. Specifically, we will analyze plume 1 and plume 3 from the challenge

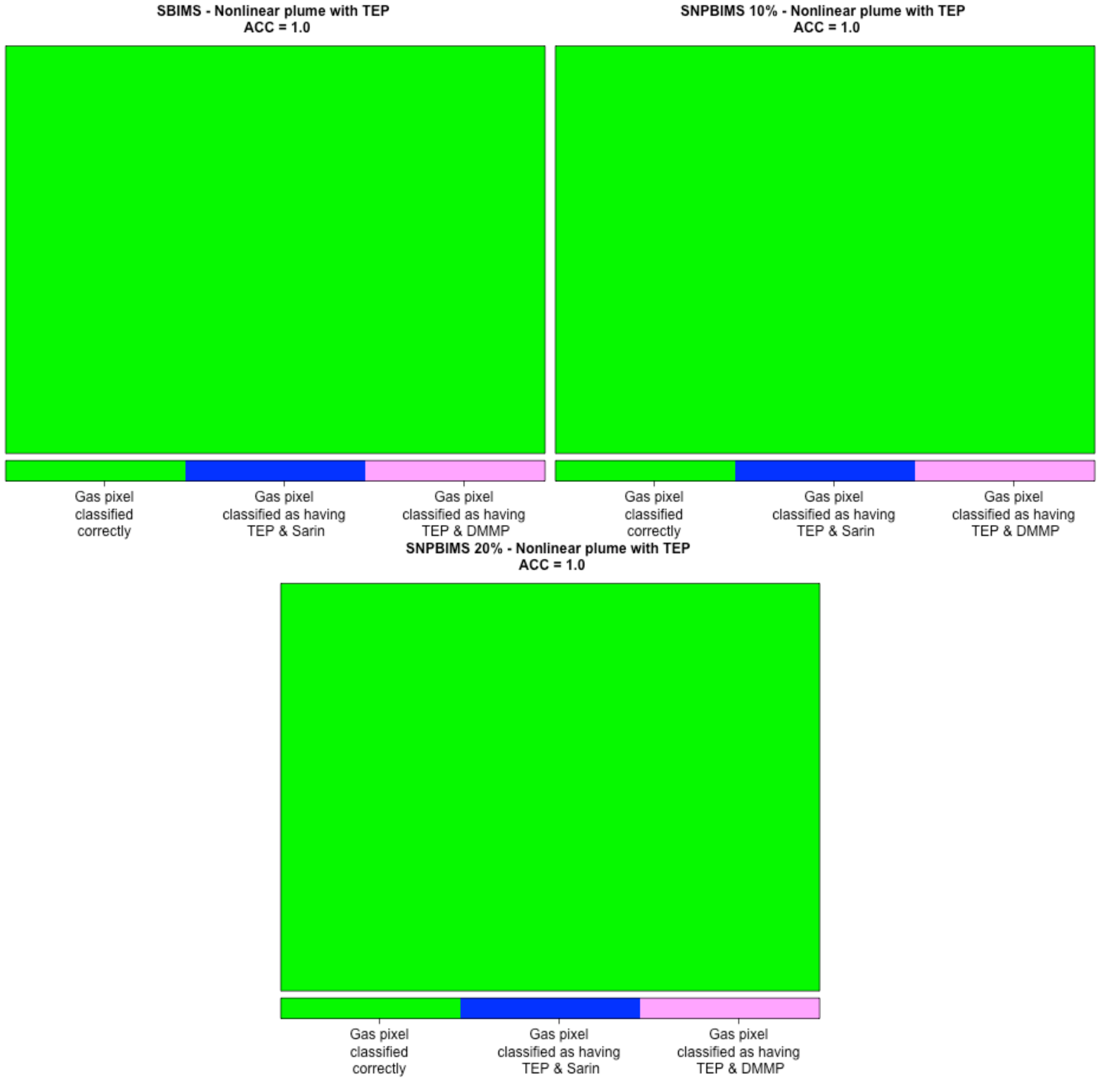


Figure 4.10: Identification results of SBIMS, SNPBIMS 10% and SNPBIMS 20% on the TEP target plume region from the nonlinear dataset.

dataset, where plume 1 contains chemicals 1 and 3, which have low correlation, and plume 3 contains chemicals 7 and 8, which are highly correlated (see Section 1.5.2). Recall that for the challenge dataset we consider 8 possible chemicals, and we will do the same for the target plume regions. Also like the previous section, since we are analyzing target plume regions and a perfect first pass detection was performed, we can use uncontaminated estimates of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, and we can give the no chemical model zero probability in the prior.

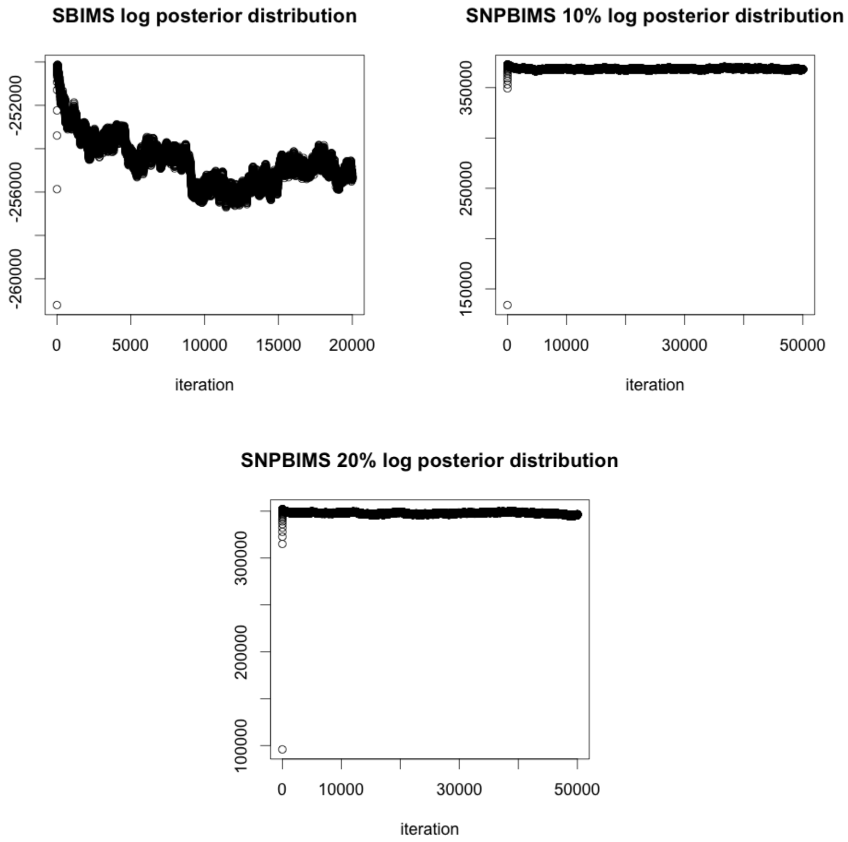


Figure 4.11: Trace plots for SBIMS, SNPBIMS 10%, and SNPBIMS 20% on the TEP target plume region from the nonlinear dataset.

4.5.1 NPBIMS on challenge dataset plume 1

In Figure 4.12 we show the identification results of BIMS and 3 versions of NPBIMS. NPBIMS 1% assumes in the prior that nonlinearities can explain 1% of the pixel signal variance, and specifically for the parameters for the IG prior on κ_j we choose $a_\kappa = 11$ and $b_\kappa = 0.1$. BIMS attains an ACC of .5109, meaning it is able to correctly classify about 51% of pixels in plume 1 correctly. On the other hand, NPBIMS 1% performs poorly since it classifies zero pixels correctly. Looking at Table 4.1 we can see that BIMS identifies two chemicals or one chemical in most pixels, however NPBIMS 1% identifies most pixels as having either 1 chemical, or all 8 chemicals. Furthermore looking at Table 4.2, we can see that BIMS identifies relatively more pixels as having chemical 1 and chemical 3 as compared to the other chemicals. In contrast, NPBIMS is almost uniformly finding all chemicals over all the pixels in the plume. The poor result of NPBIMS 1% suggests that the pixel signals in plume 1 of the challenge dataset are truly linear, i.e., no amount of nonlinearity can help explain the variance in the pixel signals. As a sanity check we test the performance of NPBIMS as we push the prior mean of κ_j to zero. NPBIMS 1^{-6}% assumes 0.000001% of the variance can be explained by non-linearities, where we set $a_\kappa = 10,001$ and $b_\kappa = 0.0001$. Lastly NPBIMS 1^{-13}% assumes 0.0000000000001% of the variance can be explained by non-linearities and we assign $a_\kappa = 100,000,001$ and $b_\kappa = 0.0000001$. We can see improvement with NPBIMS 1^{-6}%, and the result of NPBIMS 1^{-13}% is essentially the same as BIMS. Also the distribution of pixel counts in Tables 4.1 and 4.2 become closer to the BIMS results as the NPBIMS prior mean on κ_j gets closer to zero.

	Number of Chemicals							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
BIMS	383	623	72	14	5	0	0	5
NPBIMS 1%	335	24	2	1	0	0	2	738
NPBIMS 1^{-6}%	332	435	63	13	1	0	0	258
NPBIMS 1^{-13}%	428	596	38	11	0	0	0	29
Truth	0	1102	0	0	0	0	0	0

Table 4.1: Tally of pixels by number of chemicals identified for each model. For example, BIMS identified 383 pixels as containing only 1 chemical.

	Chemical							
	c1	**c2**	**c3**	**c4**	**c5**	**c6**	**c7**	**c8**
BIMS	746	50	849	51	72	108	50	40
NPBIMS 1%	740	746	740	781	783	940	784	797
NPBIMS 1^{-6}%	724	274	841	300	333	414	316	310
NPBIMS 1^{-13}%	763	68	814	52	77	118	57	61
Truth	1102	0	1102	0	0	0	0	0

Table 4.2: Tally of pixels by chemical type for each model. For example, BIMS identified 746 pixels as containing chemical 1.

We ran BIMS for 10,000 iterations with a burn in of 5,000 iterations for inference, which took about 30 minutes or .18 seconds per iteration to complete. The NPBIMS methods took about 16 hours or about 5 seconds per iteration to complete 10,000 MCMC iterations, again with a burn in of 5,000 iterations for inference. Note that plume 1 of the challenge dataset contains a total of 1,102 pixels, however 8 possible chemicals are considered for this dataset, which slows down computation compared to target plumes from the nonlinear dataset where we were only considering 3 possible chemicals. We show log posterior distribution values over the last 5,000

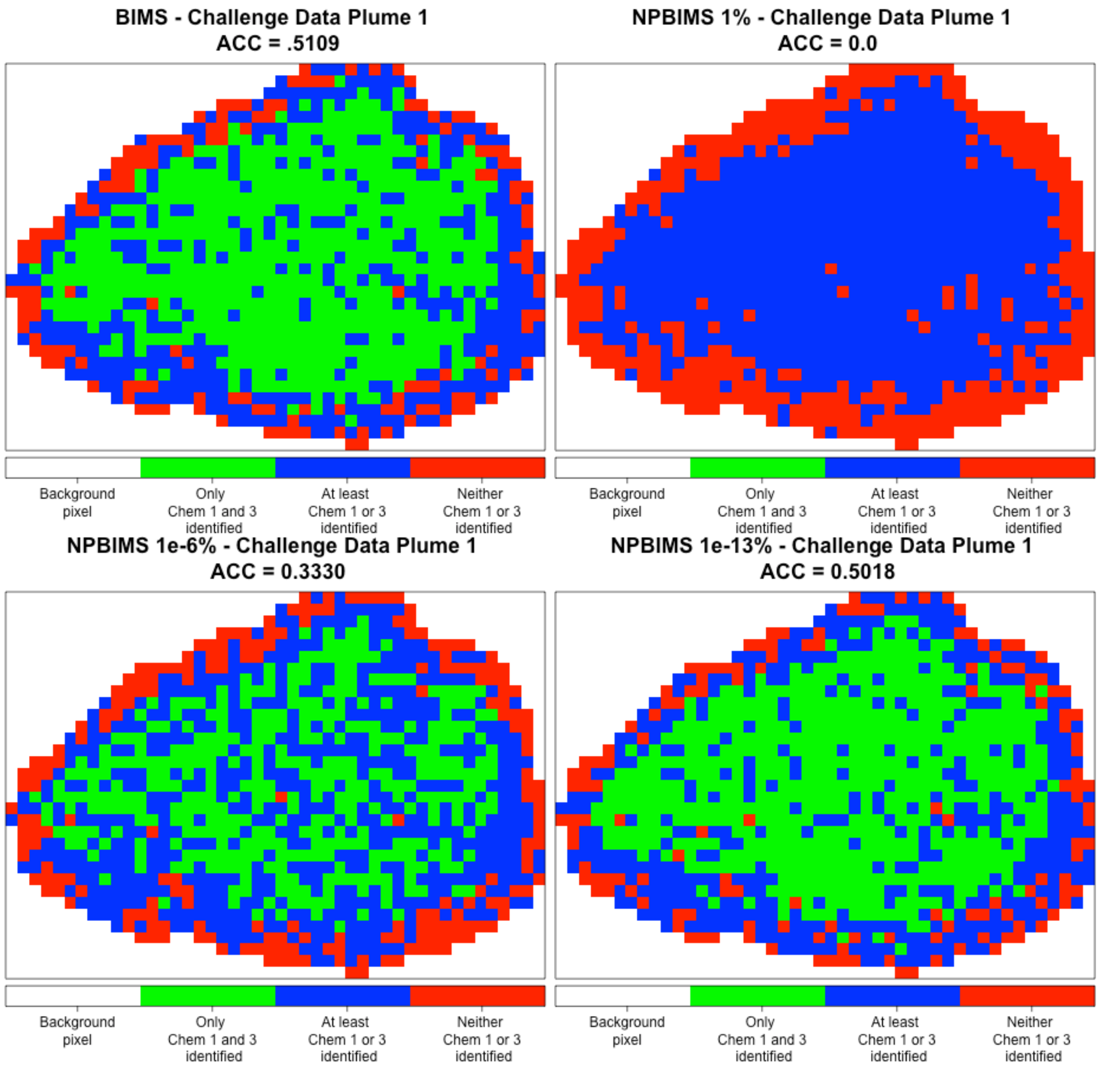


Figure 4.12: Identification results for BIMS and different versions of NPBIMS for the plume 1 target region from the challenge dataset.

iterations under each model in Figure 4.13.

4.5.2 SNPBIMS on challenge dataset plume 1

For completeness in Figure 4.14 we show results for SBIMS and SNPBIMS $1^{-13}\%$, which is the spatial version of NPBIMS $1^{-13}\%$, on plume 1 of the challenge dataset. We expect these two methods to be essentially equivalent since the prior mean on κ_j is very close to zero under SNPBIMS $1^{-13}\%$. We did not implement SNPBIMS $1^{-6}\%$ or SNPBIMS 1% on plume 1 since we found in the previous section that pixel signals in plume 1 are very linear. We can see that

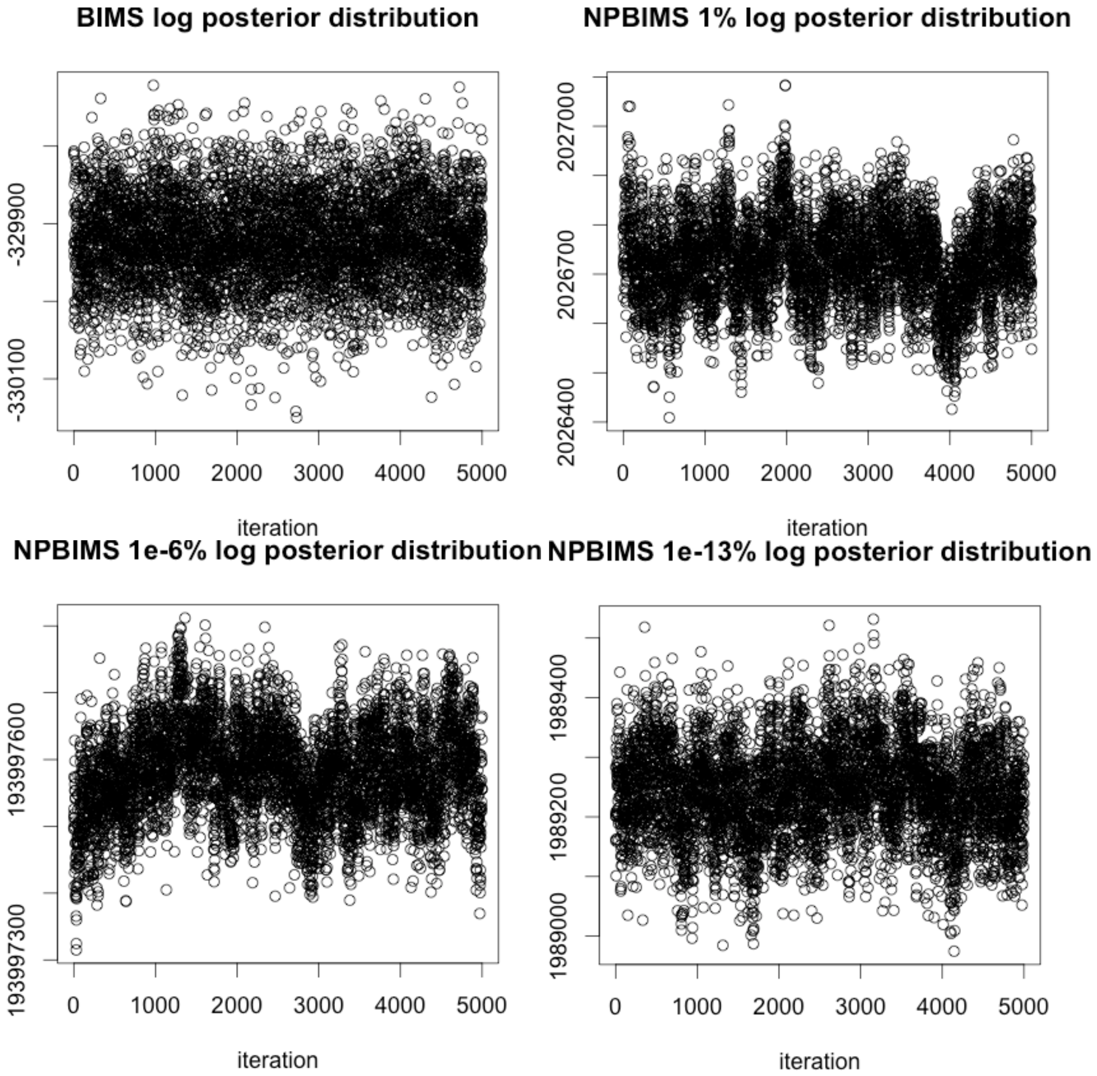


Figure 4.13: Trace plots for BIMS and different versions of NPBIMS for the plume 1 target region from the challenge dataset.

the two models yield very similar results when comparing the two plots in Figure 4.14. We can see in Table 4.3 that both methods have a smoothing effect such that only 1 or 2 chemicals are being identified in the pixels (with the exception of 1 pixel under SNPBIMS $1^{-13}\%$ that is being identified as having 3 chemicals). Furthermore looking at Table 4.4 we can see that SBIMS is identifying only the correct chemicals in the pixels, and we also see this under SNPBIMS $1^{-13}\%$ with the exception of 2 pixels.

SNPBIMS is very computationally burdensome to implement. For any version of SNPBIMS on plume 1 of the challenge dataset it takes about 6 seconds per MCMC iteration even with the

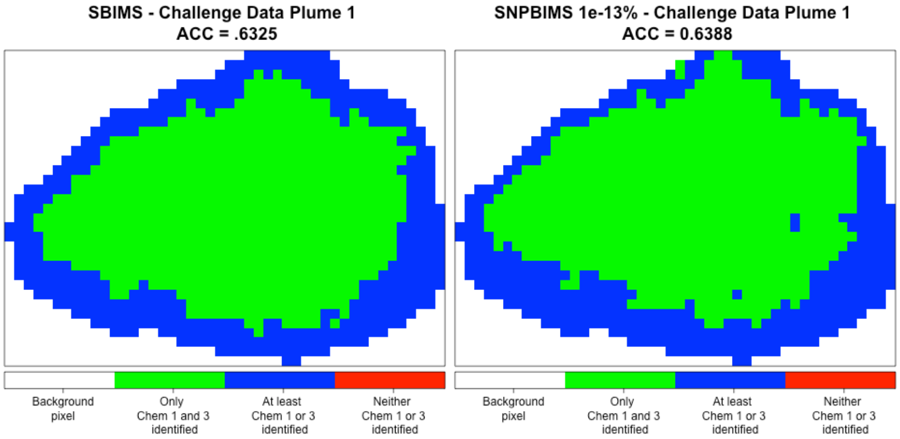


Figure 4.14: Identification results for SBIMS and SNPBIMS $1e^{-13}\%$ for the plume 1 target region from the challenge dataset.

	Number of Chemicals							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
SBIMS	405	697	0	0	0	0	0	0
SNPBIMS $1^{-13}\%$	387	714	1	0	0	0	0	0
Truth	0	1102	0	0	0	0	0	0

Table 4.3: Tally of pixels by number of chemicals identified for each model. For example, SBIMS identified 405 pixels as containing only 1 chemical.

	Chemical							
	c1	**c2**	**c3**	**c4**	**c5**	**c6**	**c7**	**c8**
SBIMS	803	0	996	0	0	0	0	0
SNPBIMS $1^{-13}\%$	816	1	1000	0	0	0	0	1
Truth	1102	0	1102	0	0	0	0	0

Table 4.4: Tally of pixels by chemical type for each model. For example, SBIMS identified 803 pixels as containing chemical 1.

use of parallel computing. For the results in Figure 4.14 we ran SNPBIMS $1^{-13}\%$ for 40,000 iterations with 30,000 burn in for inference, which took about 3 days to complete. On the other hand we ran SBIMS for 30,000 iterations with burn in of 15,000 iterations for inference, which took about 2.5 hours to complete, or .3 seconds per iteration. In the left panel of Figure 4.15 we show the log posterior distribution under SBIMS for two chains for 30,000 iterations, where we have plotted the first MCMC as blue dots and the second one as red x's. Lastly, in the right panel of Figure 4.15 we show the log posterior distribution under SNPBIMS $1^{-13}\%$ for the last 20,000 MCMC iterations.

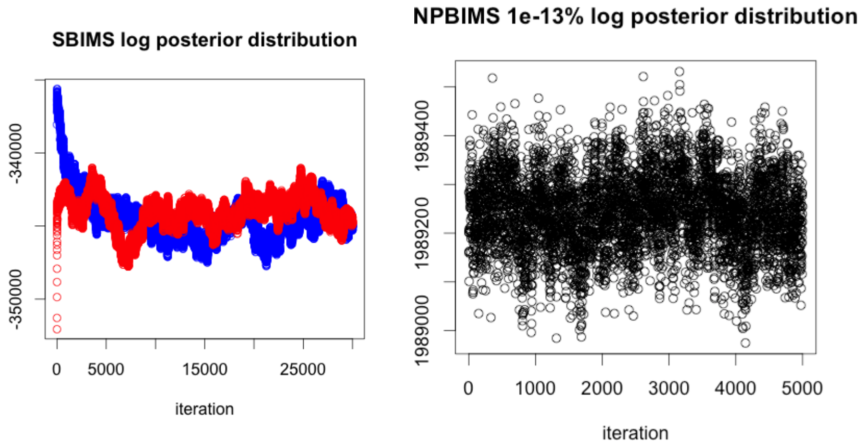


Figure 4.15: Trace plots for SBIMS and SNPBIMS $1e^{-13}$% for the plume 1 target region from the challenge dataset.

4.5.3 NPBIMS on challenge dataset plume 3

We now will discuss the performance of NPBIMS on target plume 3 of the challenge dataset. In Figure 4.16 we show the results for BIMS, NPBIMS 1%, and NPBIMS 5%. In general all methods are able to identify only a small percentage of pixels correctly, and this is because chemicals 7 and 8, which are the true chemicals present in all pixels in plume 3, are highly correlated with each other. However, NPBIMS 1% is able to correctly identify twice as many pixels than BIMS, and NPBIMS 5% has slightly better performance over NPBIMS 1%. Looking at Table 4.5 all methods are mostly identifying only one chemical in the the plume 3 pixels, even though there are actually two chemicals present. Furthermore in Table 4.5 we can see that for the majority pixels chemical 7 is identified, or chemical 8 is identified. Therefore the methods are able to identify at least one of the present chemicals.

	Number of Chemicals							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
BIMS	932	71	10	3	0	0	0	2
NPBIMS 1%	922	68	7	3	1	4	1	12
NPBIMS 5%	923	74	8	3	0	2	1	7
Truth	0	1018	0	0	0	0	0	0

Table 4.5: Tally of pixels by number of chemicals identified for each model for plume 3 target region of challenge datasets. For example, BIMS identified 982 pixels as containing only 1 chemical.

We ran BIMS on plume 3 of the challenge dataset for 10,000 iterations with burn in of 5,000

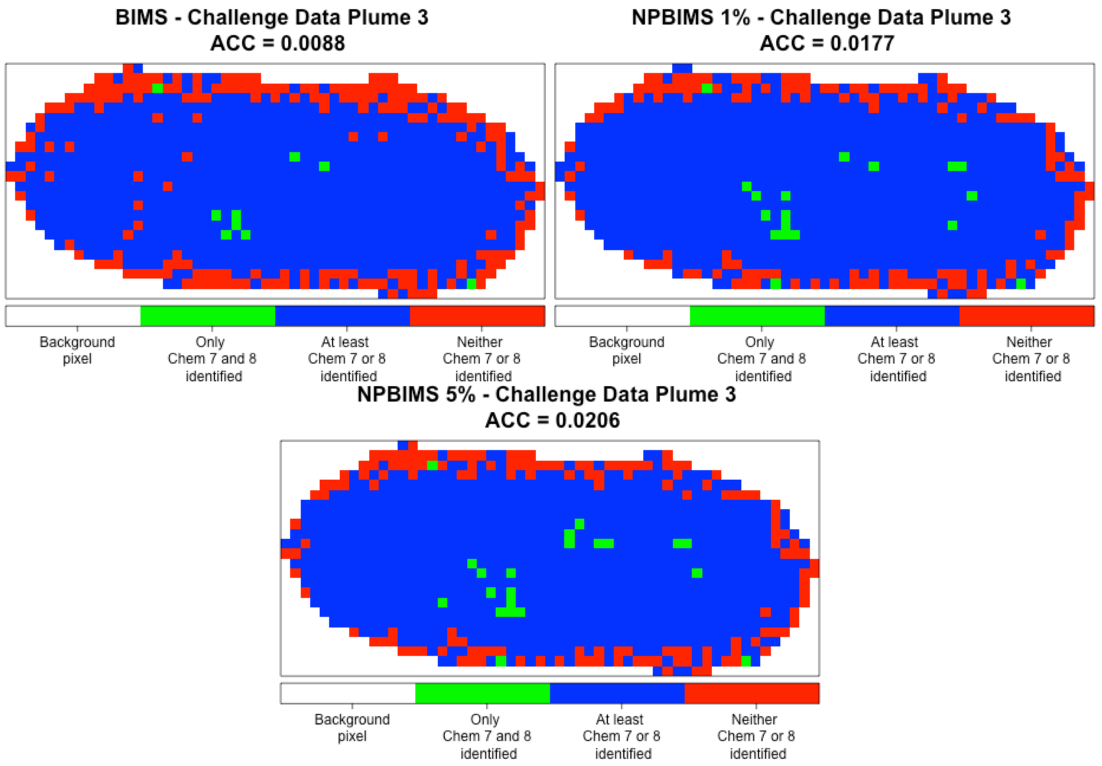


Figure 4.16: Identification results for BIMS and different versions of NPBIMS for the plume 3 target region from the challenge dataset.

	Chemical							
	c1	**c2**	**c3**	**c4**	**c5**	**c6**	**c7**	**c8**
BIMS	42	45	74	51	40	48	345	487
NPBIMS 1%	12	23	14	101	84	75	396	518
NPBIMS 5%	7	15	8	93	84	71	388	516
Truth	0	0	0	0	0	0	1018	1018

Table 4.6: Tally of pixels by chemical type for each model for plume 3 target region of challenge datasets. For example, BIMS identified 42 pixels as containing chemical 1.

iterations for inference, which took about 28 minutes to complete, or about .168 seconds per iteration. Note that plume 3 has a total of 1,018 pixels. We ran both NPBIMS 1% and NPBIMS 5% each for 10,000 iterations with 5,000 burn-in for inference. Both methods took about 14.5 hours to complete, or about 5.2 seconds per iteration. We show the log posterior distribution values under each of the three methods for the last 5,000 iterations in Figure 4.17.

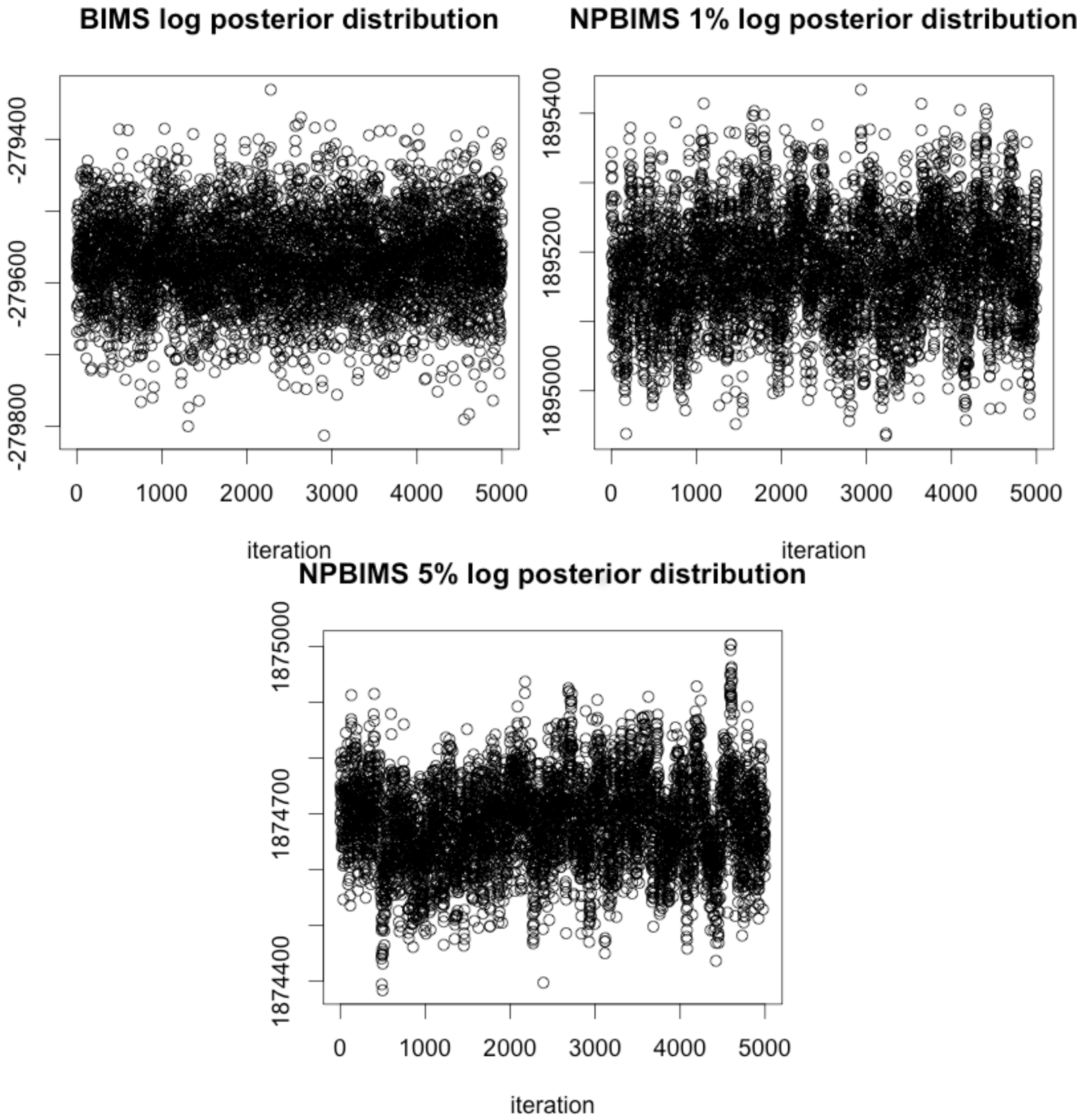


Figure 4.17: Trace plots for BIMS and different versions of NPBIMS for the plume 3 target region from the challenge dataset.

4.5.4 SNPBIMS on challenge dataset plume 3

In Figure 4.18 we present results for SBIMS, SNPBIMS 1%, and SNPBIMS 5% on plume 3 of the challenge dataset. These spatial methods perform significantly better than their nonspatial counterparts (see Section 4.5.3). Furthermore we see improvement of both SNPBIMS versions over SBIMS, with SNPBIMS 1% showing the most improvement. Examining table 4.7 we can see that pixels in plume 3 are being identified as either having 1 chemical or 2 chemicals, with the majority of pixels being identified as having only 1 chemical. Interestingly looking at Table 4.8 for the SNPBIMS methods, we can see that only chemicals 7 and 8 are being identified, and chemical 8 is identified in all pixels. Therefore both SNPBIMS models are identifying pixels correctly as having both chemicals 7 and 8, or having only chemical 8. SNPBIMS 5% is performing slightly worse than SNPBIMS 1% even though NPIMS 5% performed slightly better than NPBIMS 1%. We posit this is happening for a two reasons. First, it is the nature of the particular plume region. And secondly, the combination of accounting for spatial structure and placing a larger nonlinear percentage term, such as 5% versus 1%, is causing over smoothing.

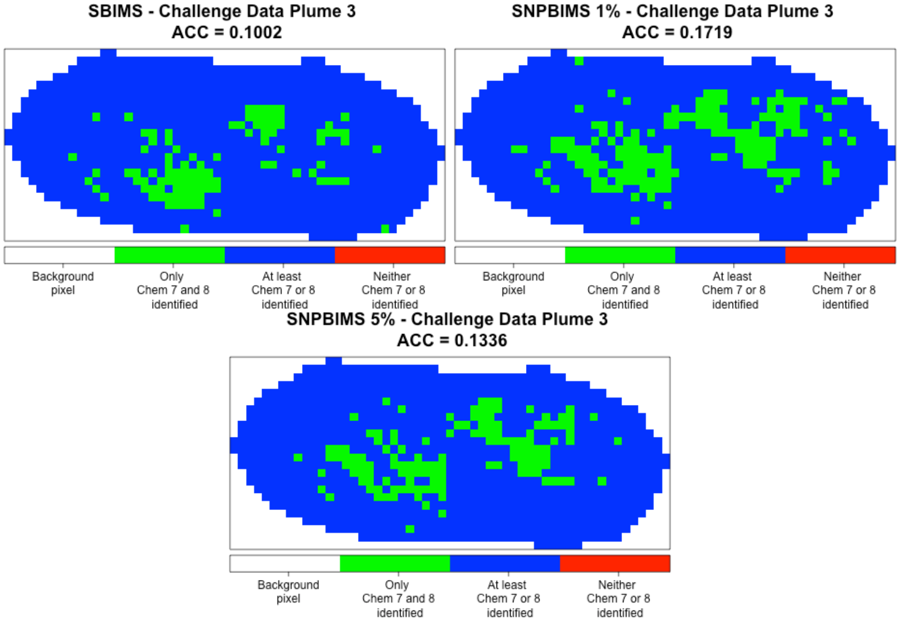

SBIMS - Challenge Data Plume 3
ACC = 0.1002
SNPBIMS 1% - Challenge Data Plume 3
ACC = 0.1719
SNPBIMS 5% - Challenge Data Plume 3
ACC = 0.1336
Background pixel
Only Chem 7 and 8 identified
At least Chem 7 or 8 identified
Neither Chem 7 or 8 identified

Figure 4.18: Identification results for SBIMS and different versions of SNPBIMS for the plume 3 target region from the challenge dataset.

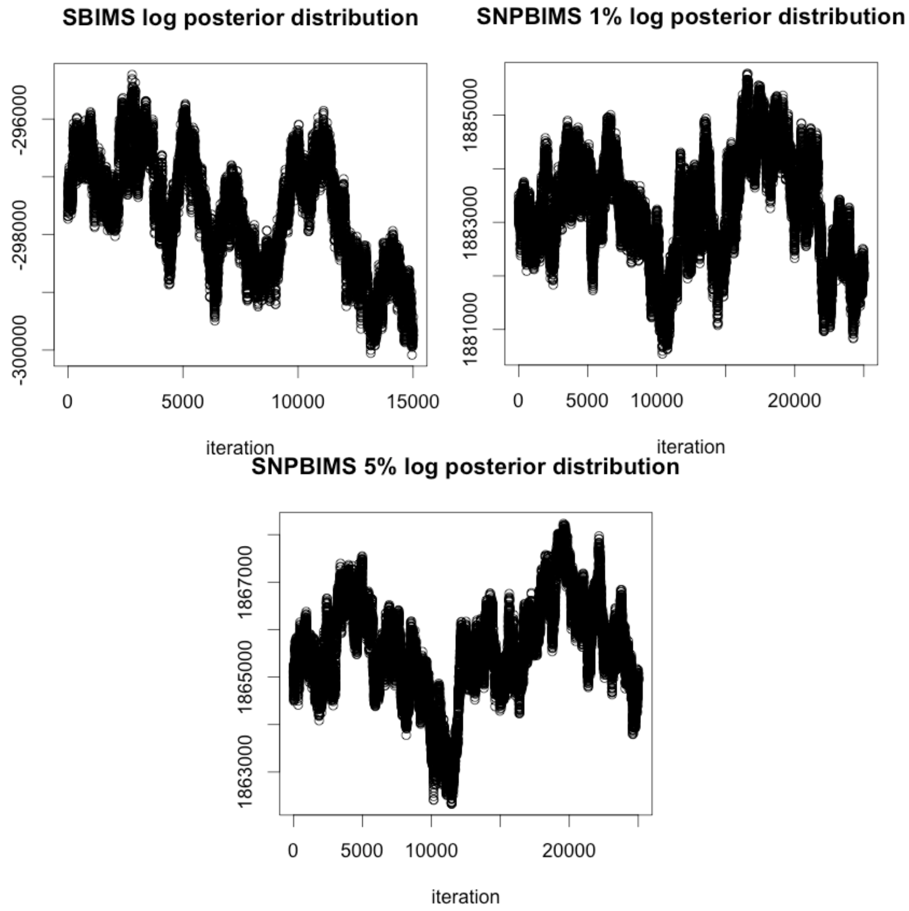
	Number of Chemicals							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
SBIMS	914	104	0	0	0	0	0	0
SNPBIMS 1%	844	174	0	0	0	0	0	0
SNPBIMS 5%	885	133	0	0	0	0	0	0
Truth	0	1018	0	0	0	0	0	0

Table 4.7: Tally of pixels by number of chemicals identified for each model for plume 3 target region of challenge datasets. For example, SBIMS identified 914 pixels as containing only 1 chemical.

We ran BIMS for 30,000 iterations with 15,000 burn-in for inference, which took about 1.8

	Chemical							
	c1	**c2**	**c3**	**c4**	**c5**	**c6**	**c7**	**c8**
SBIMS	0	1	1	0	0	0	336	784
SNPBIMS 1%	0	0	0	0	0	0	174	1018
SNPBIMS 5%	0	0	0	0	0	0	133	1018
Truth	0	0	0	0	0	0	1018	1018

Table 4.8: Tally of pixels by chemical type for each model for plume 3 target region of challenge datasets. For example, SBIMS identified 0 pixels as containing chemical 1.


SBIMS log posterior distribution

SNPBIMS 1% log posterior distribution

SNPBIMS 5% log posterior distribution

Figure 4.19: Trace plots for SBMS and different versions of SNPBIMS for the plume 3 target region from the challenge dataset.

hours to complete, or .2 seconds per iteration. Log posterior distribution values under BIMS are shown in Figure 4.19 for the last 15,000 iterations. We ran both SNPBIMS models for 50,000 iterations with 40,000 burn in for inference. Both models took a little over 3 days to complete, or about 5.25 seconds per iteration. For both SNPBIMS models we show the log posterior distribution for the last 25,000 in Figure 4.19.

4.6 Discussion

For the target plumes from the nonlinear dataset we saw that there was improvement of NPBIMS over BIMS, and this was especially true for the TEP/DMMP dataset. However we saw that because SNPBIMS takes into account nonlinearities and spatial structure it can be prone to over smoothing. We saw similar behavior for plume 3 from the challenge dataset, but for this target plume region, we actually saw some improvement of SNPBIMS over SBIMS. On the other hand, plume 1 from the challenge dataset presented a scenario where NPBIMS and SNPBIMS could at best perform only equally as well as their parametric counterparts, BIMS and SBIMS, by choosing a prior mean on κ_j that is essentially zero. Hence, plume 1 pixels from the challenge dataset are very linear and there was no benefit in taking into account nonlinearity. Although we did see improvement of NPBIMS over BIMS, SNPBIMS performed worse, equivalently, or only somewhat better than SBIMS. Therefore, we find that it is more important to take into account spatial structure in the image versus taking into account nonlinearity within pixel signals.

Chapter 5

Computational Algorithms

5.1 Computational algorithm for LASSO and FL

For the LASSO and FL discussed in Sections 2.3 and 3.1, the main goal is to infer which elements in $\boldsymbol{g}_j$ are zero and which are non-zero for each pixel, $j = 1, ..., J$. To do this we obtain the LASSO and FL estimates of $\boldsymbol{g}_j$ using the Split Bregman algorithm outlined in Ye and Xie (2011). First we outline the Split Bregman for the LASSO in Section 5.1.1, and then we do the same for FL in section 5.1.2.

5.1.1 Split Bregman for the LASSO

The Split Bregman algorithm for the LASSO first turns the unconstrained LASSO optimization problem in equation (2.4) to a constrained one,

$$\begin{aligned} &\min_{\boldsymbol{g}_j} ||\boldsymbol{y}_j - \boldsymbol{Z}\boldsymbol{g}_j||^2 + \eta_j ||\boldsymbol{g}_j||_1 \\ &\text{s.t. } \boldsymbol{a}_j = \boldsymbol{g}_j. \end{aligned} \tag{5.1}$$

The augmented Lagrangian of equation (5.1) is

$$\mathscr{L}(\boldsymbol{g}_j, \boldsymbol{a}_j, \boldsymbol{r}_j) = ||\boldsymbol{y}_j - \boldsymbol{Z}\boldsymbol{g}_j||^2 + \eta_j ||\boldsymbol{g}_j||_1 + \boldsymbol{r}_j'(\boldsymbol{g}_j - \boldsymbol{a}_j) + \frac{p}{2}||\boldsymbol{g}_j - \boldsymbol{a}_j||^2, \tag{5.2}$$

where $\boldsymbol{r}_j$ is a dual variable corresponding to the linear constraint, $\boldsymbol{a}_j = \boldsymbol{g}_j$, and the last term on the right hand side is a penalty term for violating the constraint, $\boldsymbol{a}_j = \boldsymbol{g}_j$. Note that we set $p = 1$. For details on how to choose p see Ye and Xie (2011). We then find the saddle point, $(\boldsymbol{g}_j^*, \boldsymbol{a}_j^*, \boldsymbol{r}_j^*)$ s.t. $\mathscr{L}(\boldsymbol{g}_j^*, \boldsymbol{a}_j^*, \boldsymbol{r}_j) \leq \mathscr{L}(\boldsymbol{g}_j^*, \boldsymbol{a}_j^*, \boldsymbol{r}_j^*) \leq \mathscr{L}(\boldsymbol{g}_j, \boldsymbol{a}_j, \boldsymbol{r}_j^*)$. It can be shown that $\boldsymbol{g}_j^*$ is an optimal solution for the LASSO optimization problem in equation (2.4). We solve for the saddle point $(\boldsymbol{g}_j^*, \boldsymbol{a}_j^*, \boldsymbol{r}_j^*)$ in an iterative fashion. First we choose initial values, $\boldsymbol{g}_j^0, \boldsymbol{a}_j^0, \boldsymbol{r}_j^0$, then we repeat the following steps until convergence.

1. $\boldsymbol{g}_j^{k+1} = (\boldsymbol{Z}'\boldsymbol{Z} + p\boldsymbol{I})^{-1}(\boldsymbol{Z}'\boldsymbol{y}_j - \boldsymbol{r}_j^k + p\boldsymbol{a}_j^k)$
2. $\boldsymbol{a}_j^{k+1} = \left(t_{\frac{\eta}{p}}(g_{1,j}^{k+1} + \frac{r_{m,j}}{p}), ..., t_{\frac{\eta}{p}}(g_{M,j}^{k+1} + \frac{r_{M,j}^k}{p}) \right)$
3. $\boldsymbol{r}_j^{k+1} = \boldsymbol{r}_j^k + \delta \left(\boldsymbol{g}_j^{k+1} - \boldsymbol{a}_j^{k+1} \right)$

where $t_e(q) = \text{sgn}(q) \max(0, |q| - e)$, which is the well known soft-thresholding operator. The step for updating $\boldsymbol{r}_j$ uses gradient descent with step size, $\delta = 1$. Let $O^k = ||\boldsymbol{y}_j - \boldsymbol{Z}'\boldsymbol{g}_j||^2$. The iterative algorithm above reaches convergence when $\frac{|O^{k+1} - O^k|}{|O^k|} < 10^{-4}$. Note that we implement the Split Bregman for the LASSO independently and in parallel over all J pixels.

5.1.2 Split Bregman for the FL

Let $\boldsymbol{Y}$ be an $I \times J$ matrix where the columns of $\boldsymbol{Y}$ are $(\boldsymbol{y}_1, ..., \boldsymbol{y}_J)$. Let $\boldsymbol{G}$ be an $M \times J$ matrix whose columns are $(\boldsymbol{g}_1, ..., \boldsymbol{g}_J)$. Then we can rewrite the FL optimization problem in equation (3.1) as

$$\min_{\boldsymbol{G}} tr(\boldsymbol{Y} - \boldsymbol{Z}\boldsymbol{G})'(\boldsymbol{Y} - \boldsymbol{Z}\boldsymbol{G}) + \xi||\boldsymbol{G}||_1 + \zeta||\boldsymbol{G}\boldsymbol{N}||_1 \tag{5.3}$$

where if $\boldsymbol{N}$ is an $C \times D$ matrix, then $||\boldsymbol{N}||_1 = \sum_{c=1}^{C} \sum_{d=1}^{D} |N_{c,d}|$. $\boldsymbol{G}\boldsymbol{N}$ rewrites the term $\sum_{j=1}^{J} \sum_{j'=1, j' \neq j}^{J} \sum_{m=1}^{M} w_{j,j'} \, |g_{m,j} - g_{m,j'}|$ from the FL optimization problem in equation (3.1). $\boldsymbol{N}$ has dimension $M \times Q$ and it defines a first order neighborhood among the differences of neighboring regression vectors, $\boldsymbol{g}_1, ...\boldsymbol{g}_J$. Similar to Section 5.1.1, we can rewrite the optimization in equation (5.3) as a constrained optimization problem by adding constraints, $\boldsymbol{A} = \boldsymbol{G}$ and $\boldsymbol{B} = \boldsymbol{G}\boldsymbol{N}$. The augmented Lagrangian of the constrained FL optimization problem is,

$$\mathscr{L}(\boldsymbol{G}, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{R}, \boldsymbol{V}) = tr(\boldsymbol{Y} - \boldsymbol{Z}\boldsymbol{G})'(\boldsymbol{Y} - \boldsymbol{Z}\boldsymbol{G}) + tr\boldsymbol{R}'(\boldsymbol{G} - \boldsymbol{A}) + tr\boldsymbol{V}'(\boldsymbol{B}\boldsymbol{N} - \boldsymbol{G})$$

$$+\frac{p_1}{2}tr(\boldsymbol{G}-\boldsymbol{A})'(\boldsymbol{G}-\boldsymbol{A})+\frac{p_2}{2}tr(\boldsymbol{G}\boldsymbol{N}-\boldsymbol{B})'(\boldsymbol{G}\boldsymbol{N}-\boldsymbol{B}), \tag{5.4}$$

where $\boldsymbol{R}$ and $\boldsymbol{V}$ are dual variables corresponding to the constraints, $\boldsymbol{A} = \boldsymbol{G}$, and $\boldsymbol{B} = \boldsymbol{G}\boldsymbol{N}$, respectively, and the last two terms on the right hand side are penalties for violating these constraints. In can be shown that $\boldsymbol{G}^*$ in the saddle point $(\boldsymbol{G}^*, \boldsymbol{A}^*, \boldsymbol{B}^*, \boldsymbol{R}^*, \boldsymbol{V}^*)$ such that $\mathscr{L}(\boldsymbol{G}^*, \boldsymbol{A}^*, \boldsymbol{B}^*, \boldsymbol{R}, \boldsymbol{V}) \leq \mathscr{L}(\boldsymbol{G}^*, \boldsymbol{A}^*, \boldsymbol{B}^*, \boldsymbol{R}^*, \boldsymbol{V}^*) \leq \mathscr{L}(\boldsymbol{G}, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{R}^*, \boldsymbol{V}^*)$ is the optimal solution to the FL optimization problem in equation (5.3). Again, we solve for the saddle point in a recursive fashion. First we initialize, $\boldsymbol{G}^0, \boldsymbol{A}^0, \boldsymbol{B}^0, \boldsymbol{R}^0, \boldsymbol{V}^0$, then repeat the following steps until convergence. Set, $\boldsymbol{G}^{k+1} = \boldsymbol{G}^k$ then,

1. $G^{k+1}_{m,j} = \frac{\boldsymbol{Y}'_j\boldsymbol{Z}_m - \boldsymbol{G}^{k+1}_{j,(-m)}\boldsymbol{Z}'_{-m}\boldsymbol{Z}_m - R^k_{m,j} - \boldsymbol{N}_j\boldsymbol{V}'^k_m + p_1 A^k_{m,j} - p_2\boldsymbol{G}^{k+1}_{m,(-j)}\boldsymbol{N}_{(-j)}\boldsymbol{N}'_j + p_2\boldsymbol{N}_j\boldsymbol{B}'^k_m}{\boldsymbol{Z}'_m\boldsymbol{Z}_m + p_1 + p_2\boldsymbol{N}_j\boldsymbol{N}_{-j}}$

2. $A^{k+1}_{m,j} = t_{\frac{\eta}{p_1}}(G^{k+1}_{m,j} + \frac{R^k_{m,j}}{p_1})$

3. $B^{k+1}_{m,q} = t_{\frac{\eta_2}{p_2}}((\boldsymbol{G}^{k+1}\boldsymbol{N})_{m,q} + \frac{V^k_{m,q}}{p_2})$

4. $R^{k+1}_{m,j} = R^k_{m,j} + \delta_1(G^{k+1}_{m,j} - A^{k+1}_{m,j})$

5. $V^{k+1}_{m,q} = V^k_{m,q} + \delta_2((\boldsymbol{G}^{k+1}\boldsymbol{N})_{m,q} - B^{k+1}_{m,q})$

Convergence is reached when $\max\left|\boldsymbol{G}^{k+1} - \boldsymbol{G}^k\right| \leq 10^{-12}$. Note that step 1 can be repeated until another convergence value is reached before moving on to step 2. This results in faster convergence.

5.2 Bayesian MCMC algorithms

For detection and identification purposes we are interested in the marginal posterior distribution of the indicator vectors, $\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_J$. In particular, chemical m is declared to be present in pixel j if $\Pr(\omega_{m,j} = 1 \mid \boldsymbol{y}_1, \ldots, \boldsymbol{y}_J)$ is above a given threshold (usually, 1/2). The model that includes covariates that satisfy this condition is sometimes called the "median probability model" in the literature (Barbieri and Berger, 2004). Note, however, that no closed-form expression is available for this marginal posterior distribution of interest. Furthermore, although this posterior distribution lives in a finite discrete space, the number of possible configurations is exponentially large (more specifically, 2^M). Hence, we proceed to explore $p(\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_J \mid \boldsymbol{y}_1, \ldots, \boldsymbol{y}_J)$

using Markov chain Monte Carlo algorithms (Robert and Casella, 2005). The algorithm alternates between sampling from the full conditional distribution of each $\omega_{m,j}$ given the rest of the parameters using a random-walk Metropolis Hastings algorithm, and sampling from the full conditional distributions of all other parameters given the indicators $\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_J$. Details of the algorithms for each Bayesian method, namely BIMS, NPBIMS, and their spatial extensions, are provided in the following sections.

5.2.1 BIMS Full Conditional Distributions

- For the mixing parameters of the hyper-g priors:

$$\begin{aligned} p(c_j|\cdots) &\propto (c_j+1)^{-M(\boldsymbol{\omega}_j)/2} \\ &\left\{\boldsymbol{y}_j'\left(I - \frac{c_j}{c_j+1} Z'(\boldsymbol{\omega}_j)\{Z'(\boldsymbol{\omega}_j)Z(\boldsymbol{\omega}_j)\}^{-1}Z(\boldsymbol{\omega}_j)\right)\boldsymbol{y}_j\right\}^{-I/2} c_j^{-1.5}\exp\{-(.5I)/c_j\} \end{aligned}$$

- For the gas indicators:

$$\begin{aligned} p(\boldsymbol{\omega}_j|\cdots) \propto (c_j+1)^{-M(\boldsymbol{\omega}_j)/2} &\left\{\boldsymbol{y}_j'\left(I - \frac{c_j}{c_j+1} Z'(\boldsymbol{\omega}_j)\{Z'(\boldsymbol{\omega}_j)Z(\boldsymbol{\omega}_j)\}^{-1}Z(\boldsymbol{\omega}_j)\right)\boldsymbol{y}_j\right\}^{-I/2} \\ &\Gamma\left(1 + \sum_{m=1}^{M}\omega_{m,j}\right)\Gamma\left(1 + M - \sum_{m=1}^{M}\omega_{m,j}\right) \end{aligned}$$

5.2.2 SBIMS Full Conditional Distributions

- For the mixing parameters of the hyper-g priors:

$$\begin{aligned} p(c_j|\cdots) &\propto (c_j+1)^{-M(\boldsymbol{\omega}_j)/2} \\ &\left\{\boldsymbol{y}_j'\left(I - \frac{c_j}{c_j+1} Z'(\boldsymbol{\omega}_j)\{Z'(\boldsymbol{\omega}_j)Z(\boldsymbol{\omega}_j)\}^{-1}Z(\boldsymbol{\omega}_j)\right)\boldsymbol{y}_j\right\}^{-I/2} c_j^{-1.5}\exp(-(.5I)/c_j) \end{aligned}$$

- For the joint of the latent cGMRFs and the gas indicators can be factorized as:

$$\begin{aligned} p(\omega_{m,j}|\cdots) \propto (c_j+1)^{-M(\omega_j)/2} &\left(\boldsymbol{y}_j'\left(I - \frac{c_j}{c_j+1} Z'(\omega_j)\{Z'(\omega_j)Z(\omega_j)\}^{-1}Z(\omega_j)\right)\boldsymbol{y}_j\right)^{-I/2} \\ \Phi&\left(\sqrt{u_m\sum_{j'\neq j}\upsilon_{j,j'}}\left[\{1-\rho_m\}\alpha_m + \rho_m\frac{\sum_{j'\neq j}\upsilon_{j,j'}\ell_{m,j}}{\sum_{j'\neq j}\upsilon_{j,j'}}\right]\right)^{\omega_{m,j}} \end{aligned}$$

$$\left\{1-\Phi\left(\sqrt{u_m\sum_{j'\neq j} w_{j,j'}}\left[\{1-\rho_m\}\,\alpha_m+\rho_m\frac{\sum_{j'\neq j} w_{j,j'}\ell_{m,j}}{\sum_{j'\neq j} w_{j,j'}}\right]\right)\right\}^{1-\omega_{m,j}}$$

and

$$p(\ell_{m,j}\mid\omega_{m,j},\cdots)\propto\begin{cases}\exp\left(-\frac{u_m}{2}(\boldsymbol{\ell}_m-\alpha_m\mathbf{1}_J)'(\boldsymbol{D}_\Upsilon-\rho_m\boldsymbol{\Upsilon})(\boldsymbol{\ell}_m-\alpha_m\mathbf{1}_J)\right)\mathbf{I}_{\ell_{m,j}\geq 0} & \omega_{m,j}=1\\ \exp\left(-\frac{u_m}{2}(\boldsymbol{\ell}_m-\alpha_m\mathbf{1}_J)'(\boldsymbol{D}_\Upsilon-\rho_m\boldsymbol{\Upsilon})(\boldsymbol{\ell}_m-\alpha_m\mathbf{1}_J)\right)\mathbf{I}_{\ell_{m,j}<0} & \omega_{m,j}=0\end{cases}$$

- For the means of the cGMRFs:

$$p(\alpha_m|\cdots)\propto\exp\left(-\frac{u_m\mathbf{1'}(\boldsymbol{D}_\Upsilon-\rho_m\boldsymbol{\Upsilon})\mathbf{1}+\tau_0}{2}\left(\alpha_m-\frac{u_m\boldsymbol{\ell}'_m(\boldsymbol{D}_\Upsilon-\rho_m\boldsymbol{\Upsilon})\mathbf{1}+\tau_0\alpha_0}{u_m\mathbf{1'}(\boldsymbol{D}_\Upsilon-\rho_m\boldsymbol{\Upsilon})\mathbf{1}+\tau_0}\right)^2\right)$$

- For the precisions of the cGMRFs:

$$p(u_m|\cdots)\propto u_m^{a_u+J/2-1}\exp\left(-\left(b_u+\frac{1}{2}(\boldsymbol{\ell}_m-\alpha_m\mathbf{1}_J)'(\boldsymbol{D}_\Upsilon-\rho_m\boldsymbol{\Upsilon})(\boldsymbol{\ell}_m-\alpha_m\mathbf{1}_J)\right)u_m\right)$$

- For the spatial autocorrelation of the cGMRFs:

$$p(\rho_m|\cdots)\propto|\boldsymbol{D}_\Upsilon-\rho_m\boldsymbol{\Upsilon}|^{1/2}\exp\left(-\frac{u_m}{2}(\boldsymbol{\ell}_m-\alpha_m\mathbf{1}_J)'(\boldsymbol{D}_\Upsilon-\rho_m\boldsymbol{\Upsilon})(\boldsymbol{\ell}_m-\alpha_m\mathbf{1}_J)\right)$$

5.2.3 NPBIMS Full Conditional Distributions

With the addition of the GP prior, for NPBIMS we must sample the gas indicators, g-prior mixing parameters, plus the nonlinear variance term, κ_j, and the GP covariance term, ψ_j. We borrow ideas from Savitsky et al. (2011) for MCMC sampling strategies under NPBIMS. Recall that $\boldsymbol{L}=\boldsymbol{L}(\kappa_j,\boldsymbol{\psi}_j,c_j,\boldsymbol{\omega}_j)=\hat{\Sigma}+\kappa_j C(\boldsymbol{\psi}_j,\boldsymbol{\omega}_j)+c_j\boldsymbol{S}(\boldsymbol{\omega}_j)\left(\boldsymbol{S}'(\boldsymbol{\omega}_j)\hat{\Sigma}^{-1}\boldsymbol{S}(\boldsymbol{\omega}_j)\right)^{-1}\boldsymbol{S}'(\boldsymbol{\omega}_j)$.

- We sample the gas indicator parameters and the GP covariance function parameters jointly.

$$p(\omega_j,\psi_j|....)\propto|\boldsymbol{L}|^{-1/2}\left((\boldsymbol{x}_j-\hat{\boldsymbol{\mu}})'\boldsymbol{L}^{-1}(\boldsymbol{x}_j-\hat{\boldsymbol{\mu}})\right)^{-I/2}\Gamma\left(1+\sum_m\omega_{m,j}\right)\Gamma\left(1+M-\sum_m\omega_{m,j}\right)$$

- Then we sample the g-prior mixing parameter and the non-linear variance parameter joinlty.

$$p(c_j,\kappa_j|...) \propto |\boldsymbol{L}|^{-1/2}\left((\boldsymbol{x}_j-\hat{\boldsymbol{\mu}})'\boldsymbol{L}^{-1}(\boldsymbol{x}_j-\hat{\boldsymbol{\mu}})\right)^{-I/2}\kappa_j^{-12}\exp(-0.1/\kappa_j)c_j^{-I-1}\exp(-.5I/c_j)$$

5.2.4 SNPBIMS Full Conditional Distirbutions

Under SNPBIMS we note that we are implicitly performing Reversible Jump MCMC.

- The g-prior mixing parameter and the non-linear variance parameter are sampled jointly.

$$p(c_j,\kappa_j|...) \propto |\boldsymbol{L}|^{-1/2}\left((\boldsymbol{x}_j-\hat{\boldsymbol{\mu}})'\boldsymbol{L}^{-1}(\boldsymbol{x}_j-\hat{\boldsymbol{\mu}})\right)^{-I/2}\kappa_j^{-12}\exp(-0.1/\kappa_j)c_j^{-I-1}\exp(-.5I/c_j)$$

- The joint of the latent cGMRFs, the gas indicators, and the GP covariance parameters can be factorized as:

$$\begin{aligned}p(\omega_{m,j},\psi_{m,j}|\cdots) \propto{}& |\boldsymbol{L}|^{-1/2}\left((\boldsymbol{x}_j-\hat{\boldsymbol{\mu}})'\boldsymbol{L}^{-1}(\boldsymbol{x}_j-\hat{\boldsymbol{\mu}})\right)^{-I/2}\\ &\Phi\left(\sqrt{u_m\sum_{j'\neq j}\upsilon_{j,j'}}\left[\{1-\rho_m\}\alpha_m+\rho_m\frac{\sum_{j'\neq j}\upsilon_{j,j'}\ell_{m,j}}{\sum_{j'\neq j}\upsilon_{j,j'}}\right]\right)^{\omega_{m,j}}\\ &\left\{1-\Phi\left(\sqrt{u_m\sum_{j'\neq j}\upsilon_{j,j'}}\left[\{1-\rho_m\}\alpha_m+\rho_m\frac{\sum_{j'\neq j}\upsilon_{j,j'}\ell_{m,j}}{\sum_{j'\neq j}\upsilon_{j,j'}}\right]\right)\right\}^{1-\omega_{m,j}}\end{aligned}$$

and

$$p(\ell_{m,j}\mid\omega_{m,j},\cdots)\propto\begin{cases}\exp\left(-\frac{u_m}{2}(\ell_m-\alpha_m\mathbf{1}_J)'(\boldsymbol{D}_\Upsilon-\rho_m\boldsymbol{\Upsilon})(\ell_m-\alpha_m\mathbf{1}_J)\right)\mathbf{I}_{\ell_{m,j}\geq 0} & \omega_{m,j}=1,\\ & \psi_{m,j}<1\\ \exp\left(-\frac{u_m}{2}(\ell_m-\alpha_m\mathbf{1}_J)'(\boldsymbol{D}_\Upsilon-\rho_m\boldsymbol{\Upsilon})(\ell_m-\alpha_m\mathbf{1}_J)\right)\mathbf{I}_{\ell_{m,j}<0} & \omega_{m,j}=0,\\ & \psi_{m,j}=1\end{cases}$$

- For the means of the cGMRFs:

$$p(\alpha_m|\cdots) \propto \exp\left(-\frac{u_m\boldsymbol{1'}(\boldsymbol{D}_\Upsilon - \rho_m\boldsymbol{\Upsilon})\mathbf{1}+\tau_0}{2}\left(\alpha_m - \frac{u_m\ell'_m(\boldsymbol{D}_\Upsilon-\rho_m\boldsymbol{\Upsilon})\mathbf{1}+\tau_0\alpha_0}{u_m\boldsymbol{1'}(\boldsymbol{D}_\Upsilon-\rho_m\boldsymbol{\Upsilon})\mathbf{1}+\tau_0}\right)^2\right)$$

- For the precisions of the cGMRFs:

$$p(u_m|\cdots) \propto u_m^{a_u+J/2-1}\exp\left(-\left(b_u+\frac{1}{2}(\ell_m-\alpha_m\mathbf{1}_J)'(\boldsymbol{D}_\Upsilon-\rho_m\boldsymbol{\Upsilon})(\ell_m-\alpha_m\mathbf{1}_J)\right)u_m\right)$$

- For the spatial autocorrelation of the cGMRFs:

$$p(\rho_m|\cdots) \propto |\boldsymbol{D}_\Upsilon-\rho_m\boldsymbol{\Upsilon}|^{1/2}\exp\left(-\frac{u_m}{2}(\ell_m-\alpha_m\mathbf{1}_J)'(\boldsymbol{D}_\Upsilon-\rho_m\boldsymbol{\Upsilon})(\ell_m-\alpha_m\mathbf{1}_J)\right)$$

Chapter 6

Conclusion

Our illustrations in this work have proven several things. First, Bayesian variable selection methods, like BIMS, outperforms penalized ordinary least squares approaches to variable selection, like the LASSO. In particular under BIMS we account for colinearity among chemical signatures, but the LASSO does not. Spatial methods, like FL and SBIMS provide more robust results than pixel-by-pixel methods, i.e., LASSO and BIMS. Spatial methods are especially beneficial when spatial dependence is imposed in the prior on chemical inclusion probabilities, e.g., SBIMS, as opposed to imposing spatial dependence on regression coefficients, which is the strategy of the FL. We saw slight improvement with the use of GP priors under NPBIMS, which accounts for nonlinearities in pixel signals. However, the benefit of using GP priors and GMRF's to account for spatial dependence is questionable. The computational burden for our methods are in general quite high. However, with the use of parallel computing we can implement the LASSO, FL, BIMS, and SBIMS in a reasonable amount of time. On the other hand for NPBIMS and SNPBIMS, parallel computing helped only to certain extent. And especially for SNPBIMS, even with parallel computing we were not able to obtain enough MCMC iterations to reach convergence within a reasonable amount of time.

6.1 Future Work

Currently we are focused on implementing popular detection methods to compare to the LASSO, FL, BIMS and SBIMS. Popular detection methods include the Mahalanobis distance, the generalized likelihood ratio test, principal component analysis, projection pursuit methods, the match filter (and its variants), and support vector machines. We are considering these methods as a suggestion by JRSS referees in response to a paper we submitted. We have employed the help of Bradyn Ward to produce results for these methods.

An obvious next step would be to implement SNPBIMS on faster machines so that we can properly interpret its results. The computational performance of NPBIMS and SNPBIMS suggest the need to explore alternative computational approaches. For example stochastic gradient Langevin methods (Welling and Teh, 2011), Hamiltonian Monte Carlo (Neal et al., 2011), and variational approximations (Salimans et al., 2015). We can also explore alternative formulations of the GP prior to improve performance. For example tapered versions of the covariance function (Furrer et al., 2006), and specifications of the GP through basis function (Wikle and Cressie, 1999).

Bibliography

Petter Abrahamsen. A review of gaussian random fields and correlation functions, 1997.

H. Akaike. Information theory and an extension of the maximum likelihood principle. In *Second International Symposium on Infomration Theory*, 1973.

James H. Albert and Siddhartha Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679, 1993.

S. Banerjee, A. E. Gelfand, and B. P. Carlin. *Hierarchical modeling and analysis for spatial data*. CRC Press, 2004.

Maria Maddalena Barbieri and Jame O. Berger. Optimal predicitve model selection. *The Annals of Statistics*, 32:870–897, 2004.

M Bartlett. A comment on d. v. lindleys statistical paradox. *Biometrika*, 44:533–534, 1957.

J. Berger and L. R. Pericchi. Objective bayesian methods for model selection: Introduction and comparison. *Model Selection*, pages 135–193, 2001.

C. Berrett and C. A. Calder. Data augmentation strategies fo the bayesian spatial probit regression model. *Computational Statistics and Data Analysis*, 56:478–490, 2012.

J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society*, pages 192–236, 1974.

José M Bioucas-Dias, Antonio Plaza, Nicolas Dobigeon, Mario Parente, Qian Du, Paul Gader, and Jocelyn Chanussot. Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches. *IEEE journal of selected topics in applied earth observations and remote sensing*, 5(2):354–379, 2012.

Luke N Brewer, James A Ohlhausen, Paul G Kotula, and Joseph R Michael. Forensic analysis of bioagents by x-ray and tof-sims hyperspectral imaging. *Forensic science international*, 179 (2-3):98–106, 2008.

Tom Burr and Nicolas Hengartner. Overview of physical models and statistical approaches for weak gaseous plume detection using passive infrared hyperspectral imagery. *Sensors*, 6(12): 1721–1750, 2006.

Tom Burr, Herb Fry, Brian McVey, Eric Sander, Joseph Cavanaugh, and Andrew Neath. Performance of variable selection methods in regression using variations of the bayesian information criterion. *Communications in Statistics?Simulation and Computation®*, 37(3):507–520, 2008.

Bradley P. Carlin and Siddhartha Chib. Bayesian model choice via markov chain monte carlo. *Journal of the Royal Statistical Society. Series B*, 57:473–484, 1995.

Sek M Chai, Antonio Gentile, Wilfredo E Lugo-Beauchamp, José L Cruz-Rivera, and D Scott Wills. Hyper-spectral image processing applications on the simd pixel processor for the digital battlefield. In *Computer Vision Beyond the Visible Spectrum: Methods and Applications, 1999.(CVBVS'99) Proceedings. IEEE Workshop on*, pages 130–138. IEEE, 1999.

Victor De Oliveira. Bayesian prediction of clipped gaussian random fields. *Computational Statistics & Data Analysis*, 34(3):299–314, 2000.

Plamen A Demirev, Andrew B Feldman, and Jeffrey S Lin. Chemical and biological weapons: current concepts for future defenses. *Johns Hopkins APL Technical Digest*, 26(4):321–333, 2005.

Olivier Eches, Nicolas Dobigeon, and Jean-Yves Tourneret. Enhancing hyperspectral image unmixing with spatial correlations. *Geoscience and Remote Sensing, IEEE Transactions on*, 49(11):4239–4247, 2011.

Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Assocation*, 96:1348–1360, 2001.

V. Farley, A. Vallieres, M. Chamberland, A. Villemaire, and J. F. Legualt. Performance of the first: a long-wave infrared hyperspectral imaging sensor. In J. C. Carrano and A. Zukauskas, editors, *Proceedings of SPIE*, volume 6398, 2006.

Christian Fischer and Ioanna Kakoulli. Multispectral and hyperspectral imaging technologies in conservation: current research and potential applications. *Studies in Conservation*, 51(sup1): 3–16, 2006.

Ronald Aylmer Fisher. *Statistical methods for research workers.* Genesis Publishing Pvt Ltd, 1925.

Reinhard Furrer, Marc G Genton, and Douglas Nychka. Covariance tapering for interpolation of large spatial datasets. *Journal of Computational and Graphical Statistics*, 15(3):502–523, 2006.

AJ Gelfand, Nahni Ravishanker, and Mark D Ecker. Point-referenced binary spatial data. *Generalized Linear Models: A Bayesian Perspective*, 5:373, 2000.

Alan E Gelfand and Penelope Vounatsou. Proper multivariate conditional autoregressive models for spatial data analysis. *Biostatistics*, 4(1):11–15, 2003.

Edward I. George and Robert E. McCulloch. Variable selection via gibbs sampling. *Journal of the Royal Statistical Society*, 88:881–889, 1993.

Tom Goldstein and Stanley Osher. The split bregman method for l1-regularized problems. *SIAM journal on imaging sciences*, 2(2):323–343, 2009.

Jan Gorodkin. Comparing two k-category assignments by a k-category correlation coefficient. *Computational biology and chemistry*, 28(5):367–374, 2004.

A. A. Gowen, C. P. O'Donnel, P. J. Cullen, G. Downey, and J. M. Frias. Hyperspectral imaging - an emerging process analytical tool for food quality and safety control. *Elsevier*, 31(4): 120–141, 2014.

Peter J. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Bimetrika*, 82(4):711–732, 1995.

Zhaohui Guo, Todd Wittman, and Stanley Osher. L1 unmixing and its application to hyperspectral image enhancement. In *SPIE Defense, Security, and Sensing*, pages 73341M–73341M. International Society for Optics and Photonics, 2009.

Prudhvi Gurram and Heesung Kwon. A full diagonal bandwidth gaussian kernel svm based ensemble learning for hyperspectral chemical plume detection. In *Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International*, pages 2804–2807. IEEE, 2010.

J. A. Hackwell, D. W. Warren, R. P. Bongiovi, S. J. Hansel, T. L. Hayhurst, D. J. Mabry, M. G. Sivjee, and J. W. Skinner. LWIR/MWIR imaging hyperspectral sensor for airborne and ground-based remote sensing. In M. R. Descour and J. M. Mooney, editors, *Proceedings of SPIE*, volume 2819, 1996.

Marian-Daniel Iordache, José M Bioucas-Dias, and Antonio Plaza. Sparse unmixing of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 49(6):2014–2039, 2011.

Marian-Daniel Iordache, José M Bioucas-Dias, and Antonio Plaza. Total variation spatial regularization for sparse hyperspectral unmixing. *Geoscience and Remote Sensing, IEEE Transactions on*, 50(11):4484–4502, 2012.

Ernest Ising. Beitrag zur theorie des ferromagnetismus. *Z. Phys.*, 31:253–258, 1925.

S.M. Kay. *Fundamentals of Statistical Signal Processing.* Prentice Hall, 1998.

Lynn Kuo and Bani Mallick. Variable selection for regression models. *Sankhya: The Indian Journal of Statistics*, 60:65–81, 1998.

S Land and J Friedman. Variable fusion: a new method of adaptive signal regression. *Manuscript*, 1996.

D. P. Landau. An introduction to Monte Carlo methods in statistical physics. In W. S. Kendall, F. Liang, and J.-S. Wang, editors, *Markov Chain Monte Carlo Methods: Innovations and Applications*, volume 7 of *Lecture Notes Series, Institute for Mathematical Sciences, National University of Singapore*, pages 53–92. World Scientific, 2005.

Jun Li, José M Bioucas-Dias, and Antonio Plaza. Spectral–spatial hyperspectral image segmentation using subspace multinomial logistic regression and markov random fields. *Geoscience and Remote Sensing, IEEE Transactions on*, 50(3):809–823, 2012.

Feng Liang, Rui Paulo, German Molina, Merlise A. Clyde, and Jim O. Berger. Mixtures of g-priors for bayesian variable selection. *Journal of the American Statistical Association*, 103: 410–423, 2008.

Crystal Linkletter, Derek Bingham, Nicholas Hengartner, David Higdon, and Kenny Q Ye. Variable selection for gaussian process models in computer experiments. *Technometrics*, 48 (4):478–490, 2006a.

Crystal Linkletter, Nicholas Hengartner Derek Bingham, David Higdon, and Kenny Q. Ye. Variable seletction for gaussian process models in computer experiments. *Technometrics*, 48: 478–490, 2006b.

Renfu Lu and Yud-Ren Chen. Hyperspectral imaging for safety inspection of food and agricultural products. In *Pathogen Detection and Remediation for Safe Eating*, volume 3544, pages 121–134. International Society for Optics and Photonics, 1999.

Anne-Katrin Mahlein. Plant disease detection by imaging sensors–parallels and specific demands for precision agriculture and plant phenotyping. *Plant Disease*, 100(2):241–251, 2016.

Anne-Katrin Mahlein, Erich-Christian Oerke, Ulrike Steiner, and Heinz-Wilhelm Dehne. Recent advances in sensing plant diseases for precision crop protection. *European Journal of Plant Pathology*, 133(1):197–209, 2012.

D. Manolakis, C. Siracusa, and G. Shaw. Hyerspectral subpixel target detection using the linear mixing model. *IEEE Transactions on Geoscience and Remote Sensing*, 39:1392–1409, 2001.

D Manolakis, R Lockwood, T Cooley, and J Jacobson. Hyperspectral detection algorithms: Use covariances or subspaces? Society of Photo-optical Instrumentation Engineers, 2009.

D. G. Manolakis, S. E. Golowich, and R. S. DiPietro. Long-wave infrared hyperspectral remote sensing of chemical clouds. *IEE Singal Processing Magazine*, 31(4):120–141, 2014.

Georges Matheron. Principles of geostatistics. *Economic geology*, 58(8):1246–1266, 1963.

David W Messinger. Gaseous plume detection in hyperspectral images: a comparison of methods. In *Proc. of SPIE Vol*, volume 5425, page 593, 2004.

Peter Muller, Giovanni Parmigiani, and Kenneth Rice. Fdr and bayesian multiple comparisons rules. 2006.

Radford M Neal. Monte carlo implementation of gaussian process models for bayesian regression and classification. *arXiv preprint physics/9701026*, 1997.

Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11), 2011.

Anthony O'Hagan and JFC Kingman. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–42, 1978.

R B O'Hara and J Sillanpaa. A review of bayesian variable selection methods: What, how and which. *International Society for Bayesian Analysis*, 4(1):85–118, 2009.

Stanley Osher, Martin Burger, Donald Goldfarb, Jinjun Xu, and Wotao Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling & Simulation*, 4(2):460–489, 2005.

Renfrey Burnard Potts. Some generalized order-disorder transformations. In *Mathematical proceedings of the cambridge philosophical society*, volume 48, pages 106–109. Cambridge Univ Press, 1952.

Carl Edward Rasmussen. Gaussian processes in machine learning. In *Advanced lectures on machine learning*, pages 63–71. Springer, 2004.

C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, second edition edition, 2005.

Cristian R Rojas and Bo Wahlberg. On change point detection using the fused lasso method. *arXiv preprint arXiv:1401.5408*, 2014.

Jerome Sacks, William J Welch, Toby J Mitchell, and Henry P Wynn. Design and analysis of computer experiments. *Statistical science*, pages 409–423, 1989.

Tim Salimans, Diederik Kingma, and Max Welling. Markov chain monte carlo and variational inference: Bridging the gap. In *International Conference on Machine Learning*, pages 1218–1226, 2015.

Terrance Savitsky, Marina Vannucci, and Naijun Sha. Variable selection for nonparametric gaussian process priors: Models and computational strategies. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 26(1):130, 2011.

Louis L Scharf and L Tood McWhorter. Adaptive matched subspace detectors and adaptive coherence estimators. In *Signals, Systems and Computers, 1996. Conference Record of the Thirtieth Asilomar Conference on*, pages 1114–1117. IEEE, 1996.

R. A. Schowengerdt. *Remote Sensing: Models and Methods for Image Processing.* Academic Press, 3 edition, 2006.

G. Schwartz. Estimatng the dimension of a model. *The Annals of Statistics*, 6:461–464, 1978.

James G. Scott and James O. Berger. An exploration of aspects of bayesian multiple testing. *J. Statist. Plann. Inference*, 136:2144–2162, 2006.

James G Scott, James O Berger, et al. Bayes and empirical-bayes multiplicity adjustment in the variable-selection problem. *The Annals of Statistics*, 38(5):2587–2619, 2010.

S. W. Sharpe, T. J. Johnson, R. L. Sams, P. M. Chu, G. C. Rhoderick, and P. A. Johnson. Gas-phase databases for quantitative infrared spectroscopy. *Applied spectroscopy*, 58(12): 1452–1461, 2004.

James Theiler and Bernard R Foy. Ec-glrt: Detecting weak plumes in non-gaussian hyperspectral clutter using an elliptically-contoured generalized likelihood ratio test. In *Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008. IEEE International*, volume 1, pages I–221. IEEE, 2008.

James Theiler, Bernard R Foy, and Andrew M Fraser. Characterizing non-gaussian clutter and detecting weak gaseous plumes in hyperspectral imagery. In *Proc. of SPIE Vol*, volume 5806, page 183, 2005.

P. S. Thenkabail, Lyon J. G., and A. Huete. *Hyperspectral remote sensing of vegetation.* CRC Press, 2012.

G. E. Thomas and K. Stamnes. *Radiative transfer in the atmosphere and ocean.* Cambridge University Press, 2002.

R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society*, 67(1):91–108, 2005a.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society. Series B (Statistical Methodology*, 67(1):91–108, 2005b.

Ryan J Tibshirani and Jonathan Taylor. Regularization paths for least squares problems with generalized ℓ_1 penalties. Technical report, 2010.

T. Visser. Infrared spectroscopy in environmental analysis. *Encyclopedia of Analytical Chemistry*, pages 1–22, 2000.

Jon Wakefield. A bayesian measure of the probability of false discovery in genetic epidemiology studies. *The American Journal of Human Genetics*, 81(2):208–227, 2007.

IS Weir and AN Pettitt. Binary probability maps using a hidden conditional autoregressive gaussian process with an application to finnish common toad data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 49(4):473–484, 2000.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 681–688, 2011.

Alice S Whittemore. A bayesian false discovery rate for multiple testing. *Journal of Applied Statistics*, 34(1):1–9, 2007.

Christopher K Wikle and Noel Cressie. A dimension-reduced approach to space-time kalman filtering. *Biometrika*, 86(4):815–829, 1999.

G. B. Ye and X. Xie. Split bregman method for large scale fused lasso. *Computational Statistics & Data Analysis*, 44(4):1552–1569, 2011.

S. J. Young, R. Johnson, and J. A. Hackwell. An in-scene method for atmospheric compensation of thermal hyperspectral data. *Journal of Geophysical Research-Atmospheres*, 107(6):2557–2570, 2002.

A. Zellner. On assessing prior distributions and bayesian regression analysis with g-prior distributions. *Studies in Bayesian Econometrics and Statistics*, 6:233–243, 1986.

L. Zhang, M. Guindani, F. Versace, and M. Vannucci. A spatio-temporal nonparametric bayesian variable selection model of fmri data for clustering correlated time courses. *NeuroImage*, 95: 162–175, 2014.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B*, 67:301–320, 2005.