Semi-Supervised Data Summarization: Using Spectral Libraries to Improve Hyperspectral Clustering

K. L. Wagstaff,[1] H. P. Shu,[2] D. Mazzoni,[1] and R. Castaño[1]

Hyperspectral imagers produce very large images, with each pixel recorded at hundreds or thousands of different wavelengths. The ability to automatically generate summaries of these data sets enables several important applications, such as quickly browsing through a large image repository or determining the best use of a limited bandwidth link (e.g., determining which images are most critical for full transmission). Clustering algorithms can be used to generate these summaries, but traditional clustering methods make decisions based only on the information contained in the data set. In contrast, we present a new method that additionally leverages existing spectral libraries to identify materials that are likely to be present in the image target area. We find that this approach simultaneously reduces runtime and produces summaries that are more relevant to science goals.

I. Introduction

The goal of this work is to produce high-quality, automatic summaries of hyperspectral data. Hyperspectral imagers collect large volumes of data, with observations at hundreds or thousands of different wavelengths. The large data size renders a thorough manual analysis difficult, expensive, and time consuming. For example, the Hyperion instrument on the Earth Orbiting-1 (EO-1) spacecraft regularly produces image cubes that are over a gigabyte in size (256 × 7000 pixels, at 220 wavelengths). Automated techniques for analyzing and summarizing these mega-data sets can provide two major benefits: (1) scientists can quickly obtain high-level views of the data contents, and (2) summaries produced onboard the spacecraft enable quick prioritization of data for transmission to make the best use of limited bandwidth.

One summarization approach is to partition the pixels from a given image into a set of k clusters. Each cluster contains pixels that are more similar to each other than to pixels in other clusters. The image can be summarized by the set of k clusters, represented by, for example, the cluster means and standard deviations (of the pixel values for each cluster). However, typical clustering algorithms are completely data driven and will produce summaries based on the strongest distinguishing factor between pixels (often,

[1] Modeling and Data Management Systems Section.

[2] Formerly a student at the California Institute of Technology, Pasadena, California, and in the Modeling and Data Management Systems Section.

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

brightness), regardless of whether or not that distinction is physically meaningful. In contrast, we seek to include knowledge from existing spectral libraries to improve the automated summaries.

In this article, we present a solution that combines the strengths of existing summarization methods with those of existing spectral libraries. We describe a knowledge-driven clustering method that incorporates laboratory spectra as "seeds" for some, or all, of the data clusters. This approach is most effective when good a priori knowledge of the scene composition is available. We contrast the summaries produced by data-driven and knowledge-driven clustering, concluding that incorporating a spectral library results in summaries with greater science value than those produced from the data alone. Knowledge-based summaries are more interpretable and more likely to be based on true compositional differences in the areas being imaged. In addition, incorporating spectral library information greatly reduces runtime, which is beneficial in any situation and critical for eventual onboard applications of these methods. We present sample results from diverse areas to illustrate the benefits of clustering with spectral libraries.

II. Summarization Using Clustering Methods

Summarization is a form of compression. In summarizing a large data set, our goal is to preserve the important, overall aspects while greatly reducing the number of bits required to represent the data. A thumbnail image, in which the original image is greatly reduced in size, is one kind of summary: it sacrifices fine detail in order to represent a much larger image in fewer bits. We aim to produce summaries of hyperspectral observations of planetary surfaces that, in contrast to a simple thumbnail image, will also contain information about the surface materials present in the hyperspectral image.

In this section, we describe summaries in more detail and describe several scenarios in which they can be used. We also explain how k-means clustering can be used to produce summaries of hyperspectral data sets, and we discuss the limitations of the basic k-means method, from a science perspective. These limitations motivate our focus on enhancing k-means clustering to incorporate existing spectral libraries when producing summaries to increase their scientific relevance.

A. Summarization Scenarios

Summaries can be used to make decisions about the relative importance, or relevance, of different data sets. Consider the following scenarios:

- An orbiting instrument, such as Hyperion, images a specific region on the Earth, stores the data onboard, and transmits a summary to scientists on the ground. If the summary suggests that the content is sufficiently interesting, the full data set can be transmitted; otherwise, it can be discarded, and the bandwidth can be used to send other data with higher importance.
- An orbiting instrument images a specific region on the Earth and generates a summary to determine if any high-priority materials are present, such as active lava or smoke plumes. This assessment can be used to prioritize data sets for transmission to the ground. This kind of analysis is already being conducted onboard EO-1, by the Autonomous Sciencecraft Experiment, to determine when to schedule follow-up imaging by Hyperion in response to volcanic events [6].
- On the ground, a scientist browses a large archive of hyperspectral images by examining their summaries. The summaries can also be used to support content-based searching and indexing, so that a user can quickly search for all images containing evidence of lava or smoke plumes, as above.

Intelligent use of bandwidth can greatly increase the science return of most missions. The ability to search and index the prohibitively large data sets produced by hyperspectral images is essential for future science investigations that rely on the information they contain.

Hyperspectral data sets present a significant challenge for summarization techniques, due to their large data volumes. Without compression, even modest image sizes, such as 256×1000 pixels, result in files that consume 0.5 megabytes per observed wavelength (using 16 bits to represent each observed value). Hyperspectral imagers commonly record scenes at hundreds or even thousands of wavelengths, resulting in massive data sets. The sheer size requires that any analysis method applied to these data sets be highly efficient.

B. Producing Hyperspectral Summaries Using *k*-Means Clustering

The k-means clustering algorithm [8] is an iterative method that partitions a data set into k distinct, well-separated groups. It can be viewed as a form of data summarization in that it reduces a data set with n items, each of dimensionality d, to a summary that contains k representative items, $k \ll n$, and an indication of which representative each original item most resembles. The amount of data is *severely* reduced, and a lot of information therefore must be discarded. The goal is to permit a focus on features of interest by preserving information that indicates where they occur in the image, even though the full spectral information for each pixel is not available in the summary. This kind of summarization is essential for situations in which an instrument can collect more data than it can return, which is the case for most, if not all, missions to other planets.

For example, an image with 256,000 pixels (256 wide by 1000 tall), observed at 500 wavelengths, consumes 244 megabytes. The same data set, summarized with 10 clusters, would require only 132 kilobytes (kB): 10 kB to describe the clusters and 122 kB to specify the cluster membership of each pixel. That is, the summary provides a compression factor of almost 1900×. Clearly, quite a bit of information is lost in the summary, just as an image thumbnail discards details. However, this summarization would allow the transmission of up to 1900 images using the bandwidth previously required for a single image. As discussed above, the most interesting images could be selected based on their summaries and then transmitted in full, at significant overall savings in bandwidth.

The procedural details of the k-means clustering algorithm [8], as applied to the task of clustering the pixels in an image, I, are as follows:

Inputs: number of clusters k, image I (collection of n pixels, each with dimensionality d)
Outputs: set of k cluster centers $\{C_i\}$ and n pixel-cluster assignments

(1) Randomly select k pixels from image I and set them to be the initial cluster centers, $C_1 \cdots C_k$. Here, C_i represents both the set of items assigned to cluster i and the center of cluster i, which is a vector of d wavelength values, $C_{i1} \cdots C_{id}$.

(2) Iterate until convergence:

  (a) Assign each of the n pixels to the most similar cluster center.

  (b) Update each cluster center C_i to represent its constituent pixels $p \in C_i$. That is, the value for C_i at wavelength w is updated as follows: $C_{iw} = (1/|C_i|) \sum_{p \in C_i} p_w$.

(3) Return the set of cluster centers $\{C_i\}$ and the pixel assignments.

As a starting point, the algorithm randomly selects k pixels from the image to be the initial k cluster centers. This initial summary of the image is likely to be very poor in quality, and it is progressively refined through the iteration of step 2. In this step, the algorithm alternates between assigning each pixel to its best-match cluster (the cluster that best represents the pixel) and updating all clusters to more closely fit their constituent pixels. Iteration continues until no pixels change their cluster membership in step 2(a). The result, in step 3, is a set of k clusters that summarize k "patterns" in the data set, as well as information about the cluster to which each pixel is assigned.

For example, consider the image shown in Fig. 1(a). This image is 220 pixels wide by 247 pixels tall. Each pixel is represented by three values (red, green, and blue intensities); the resulting file size is 156 kB. The k-means algorithm can be used to generate a highly compressed summary of this image. The parameter k can be set to any desired value; higher values of k result in finer distinctions between clusters, while lower values provide more compression. For this example, we set k to 4. In typical applications, we would analyze the same image using several different values for k. The algorithm randomly selects four pixels to be the initial cluster centers, and then assigns each pixel to the most similar of the four clusters. Figure 1(b) shows the result of the first iteration, where each pixel is represented by its cluster color (the mean color of the pixels assigned to that cluster). The randomly selected cluster centers are not well separated. Each iteration progressively refines the clusters, as shown in Figs. 1(c) through 1(e). The final result preserves much of the detail of the original but requires only a fraction of the size. We can express each pixel's cluster membership using 2 bits per pixel, and the four cluster means each require three bytes to specify. The total size of the summary is 13 kB, providing an effective compression of 12×. Hyperspectral images stand to benefit even more, with compression factors in excess of 1000×, because the multiple wavelength bands are effectively collapsed to a single feature, with colors indicating material composition.

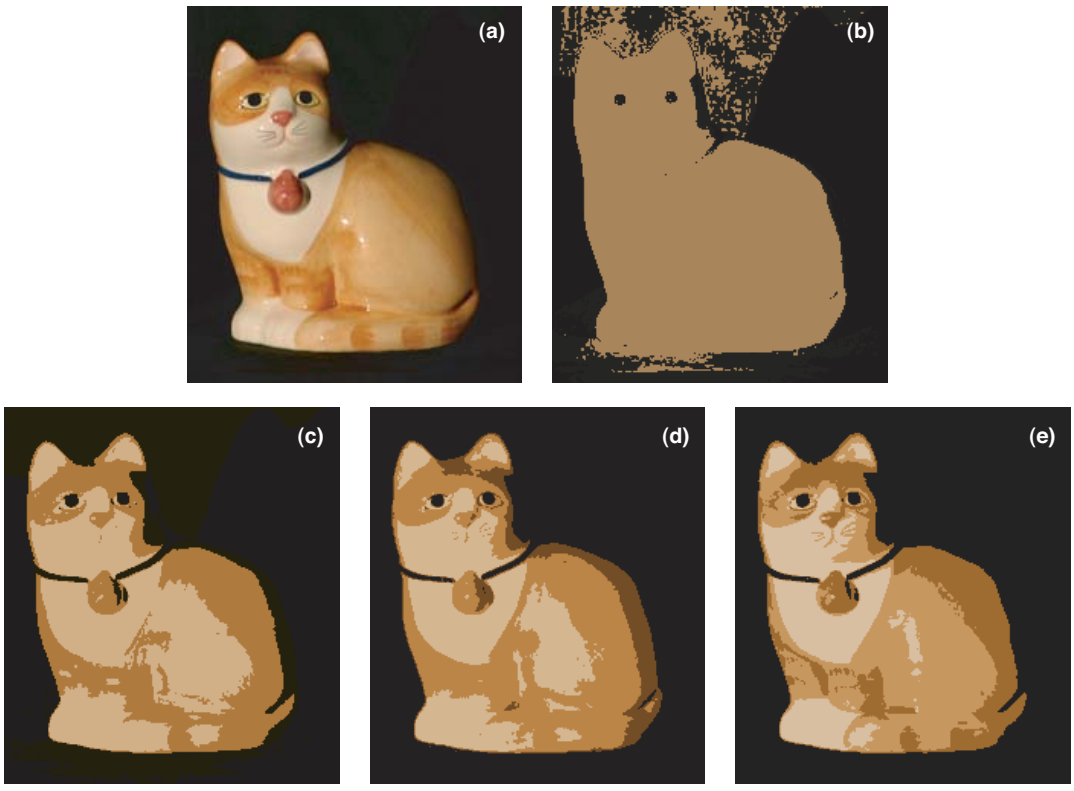

Fig. 1. The *k*-means algorithm applied to a simple image: (a) the original RGB image (156 kB) (the image is courtesy of Aaron Hertzmann and Steven Seitz); (b) iteration 1; the initial cluster centers are randomly chosen pixels, and each pixel is assigned to the most similar cluster center. The pixels are color-coded according to their cluster membership, using the mean color for their clusters. In the first iteration, image regions are not well differentiated; (c) iteration 10; (d) iteration 20; and (e) the final iteration (13 kB). In (c) through (e), each iteration progressively refines the clusters until convergence is reached. Much of the original detail is preserved, although the summary is only 1/12 the size of the original.

C. The Limitations of *k*-Means Clustering for Hyperspectral Summarization

As we have just shown, k-means clustering can provide image summaries that are much smaller than the full data set. These summaries can be used to identify the highest priority images for transmission from a remote spacecraft. They also can provide easy browse access to a large collection of images on the ground. This is particularly useful for hyperspectral images, which are difficult to quickly browse and interpret. The summaries permit a scientist to browse several images, identify the ones that contain material signatures of interest (such as water or vegetation), and then perform a more detailed analysis of just those images.

However, there are two significant shortcomings of the basic k-means algorithm that affect its utility for these scenarios. First, the speed of the algorithm is an issue. Whether it is run onboard a spacecraft to summarize newly collected data or used on the ground to process large data repositories, efficient processing is very important. The k-means algorithm has a runtime complexity that is $\mathcal{O}(nkdI)$, where

- n is the number of pixels
- k is the number of clusters
- d is the dimensionality (number of wavelength bands)
- I is the number of iterations performed by the algorithm

The number of iterations, I, is difficult to estimate prior to running the algorithm. It depends on the complexity of the image and, importantly, how inaccurate the initial cluster centers are (equivalently, how much they must change before convergence). Consequently, several researchers have investigated how best to select the initial cluster centers, such as by clustering on multiple subsamples of the data set and selecting the subsample with the best result as the initial clusters for a full k-means run [4]. Like k-means itself, this is a domain-independent solution. Our proposed solution instead leverages known properties of the data set under analysis by specifying initial cluster centers from a spectral library of common materials. As we will show, this approach provides good starting points for k-means, resulting in significantly reduced runtimes.

The second shortcoming arises from the quality of the information contained in the summaries. The basic k-means algorithm will seek the strongest separation between the k clusters, regardless of whether or not it is a sensible division. For example, a common problem with applying k-means to summarize hyperspectral images is that the resulting clusters are often separated solely by brightness. That is, the brightest (highest intensity) pixels are placed into the same cluster, and the next brightest are grouped together, and so on to the darkest pixels, which form a separate cluster. While this separation may produce a good numeric solution (in terms of minimizing overall variance), it is not necessarily physically meaningful. The bright pixel cluster could contain snow, ice, and building elements, while the dark pixel cluster could consist of a motley collection of water, shadow, and soil pixels. For this particular data analysis problem, we know that intensity alone is not the key to identifying different materials. Instead, the *shape* of the pixel's spectrum carries the most information, including relative peaks and absorption bands. Therefore, our approach seeks to overcome the default bias of the k-means algorithm (towards the biggest numeric separation) by providing spectral shape information from the spectral library. This is accomplished by "seeding" the clusters by pre-specifying their initial values with reasonable estimates of what materials may be present in the image.

III. Using Cluster Seeds to Improve *k*-Means

We have previously developed several methods for incorporating existing knowledge into the k-means clustering algorithm [9,10,12], and others have extended this work in various ways [1,3,5,7]. We have also shown how k-means can be used to analyze and summarize hyperspectral images of Mars [11]. In this

section, we describe how to "seed" the clusters to achieve the dual goals of reduced runtime and more meaningful clusters, tailored to the human analyst's goals.

The key innovation is the ability to specify what the initial cluster centers should be. These need not be exactly correct—after all, if the optimal cluster centers were fully known, we would not need to cluster at all—but they are expected to be more accurate than random selection of pixels in the image. For clustering hyperspectral images, these initial centers (seeds) come from a library that contains laboratory spectra for hundreds or thousands of known materials.

The k-means seeded clustering algorithm proceeds as follows (changes from the k-means clustering algorithm are in bold):

Inputs: number of clusters k, image I (collection of n pixels, each with dimensionality d), **initial "seed" clusters** $\{S\}$
Outputs: set of k cluster centers $\{C_i\}$ and n pixel-cluster assignments

(1) **Use $S_i \in S$ to initialize the cluster centers, $C_1 \cdots C_k$. If $|S| < k$, then randomly initialize the remaining clusters C_i.**

(2) Iterate until convergence:

   (a) Assign each of the n pixels to the most similar cluster center.

   (b) Update each cluster center C_i to represent its constituent pixels $p \in C_i$. That is, the value for C_i at wavelength w is updated as follows: $C_{iw} = (1/|C_i|) \sum_{p \in C_i} p_w$.

(3) Return the set of cluster centers $\{C_i\}$ and the pixel assignments.

This version takes an additional input S, which is the set of cluster seeds. Each S_i is a d-dimensional vector with values for each feature observed in the data set. This information is used in the modified step 1. Instead of randomly selecting pixels to be the initial cluster centers, the S_i are used to initialize the clusters. In this formulation, the user need not specify a total of k seeds; that is, it is possible for $|S| < k$. The algorithm can make use of this partial information by seeding some of the clusters and randomly initializing the rest.

To illustrate what happens when $|S| < k$, consider Fig. 2. This might happen due to incomplete knowledge of the components present in the data set or the desire to detect outliers and exceptions to known components. In this case, we have $k = 4$ but only specify seeds for three of the clusters, indicated by stars. Figure 2(a) shows a scenario in which all three seeds are good data representatives—that is, they fall within the data set domain, indicated by the cloud, showing that there is support in the data set for each of the seeds. The resulting clusters are based on the three seeds, with an additional cluster created to cover the remaining data. Thus, specifying seeds does not eliminate the ability to discover unexpected components in the data set. This is very important, since it is often the exceptions to trends, or to what is expected, that turn out to be most valuable.

Figure 2(b) addresses the issue of robustness. What if the user specifies a bad seed, one that is not relevant to the data set? If a seed falls outside the data set domain, it will end up with no items assigned to it, reflecting the fact that there is insufficient support for that seed. Thus, the seeded clustering algorithm can gracefully handle seeds that are imperfect.

Other methods exist for identifying good initial starting points for the clusters. For example, Basu et al. [1] identified cluster seeds based on a set of pairwise constraints provided by the user. When the user specifies that a subset of data points must be grouped together, that subset can be used to initialize a cluster. However, this requires that the user specify individual constraints for each data set to be analyzed. This is a reasonable general solution, if nothing more precise is known about the data set

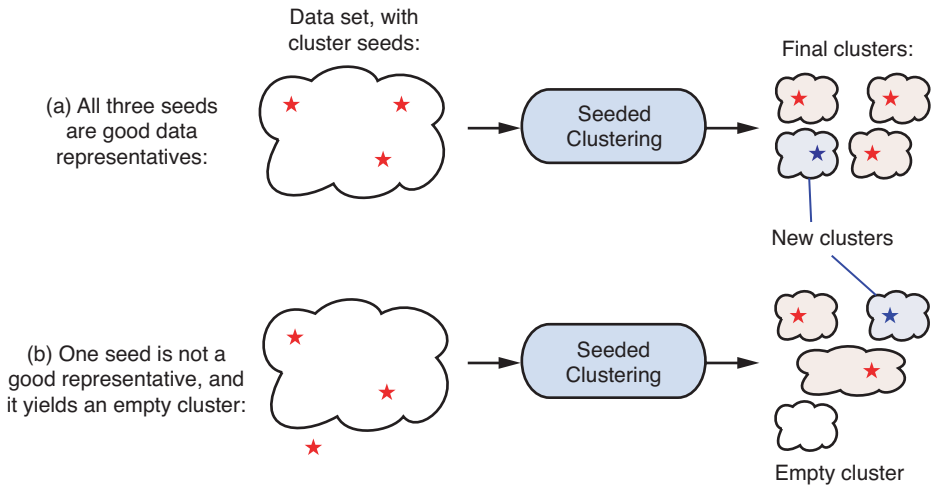

Fig. 2. Seeded clustering scenarios in which the desired number of clusters, *k*, is four, and three of the clusters are seeded (stars): (a) all three seeds have good support from items in the data set. They each end up with a cluster, and since $k = 4$, an additional cluster is created to cover the data not belonging to one of the three seeded clusters, and (b) two of the seeds are supported, and the third is not; it ends up with an empty cluster. The bottom seeded cluster spreads out to cover more of the data set, and a new cluster is created to cover the upper right part of the data set.

under study. In contrast, our solution leverages common knowledge about different materials likely to be found in hyperspectral images; the user need only select a handful of materials possibly relevant for the specific image under study, rather than hundreds or thousands of pairwise pixel decisions.

A. Spectral Libraries and Atmospheric Correction

A spectral library is a collection of spectra that were collected by observing samples of known materials with a specific instrument. They can be used for calibration and interpretation of the same instrument's later observations. In this article, we used the ASTER spectral library, which is a collection of almost 2000 spectra observed in the laboratory for use with ASTER (the Advanced Spaceborne Thermal Emission and Reflection Radiometer). This library includes data from three other spectral libraries: the Johns Hopkins University Spectral Library, the Jet Propulsion Laboratory Spectral Library, and the United States Geology Survey Spectral Library. Figure 3 shows three sample spectra that are included in the ASTER library.

Because the spectra in the ASTER library were collected under laboratory conditions, the observations we obtain from orbit may not match the spectral library very well. The raw data are affected by the atmosphere that intervenes between the orbiting hyperspectral imager and the surface being observed. Water vapor and CO_2 in the atmosphere, in particular, absorb energy at characteristic wavelengths and cause "gaps," called absorption bands, to appear in the observed spectra at those wavelengths. These gaps and other effects cause large discrepancies between the observed and laboratory spectra.

By making some reasonable assumptions about the composition of the atmosphere for each particular captured scene, it is possible to model the effects of the atmosphere on the measured spectra and correct for this, using an atmospheric band model radiation transport model such as MODTRAN [2]. Several commercial software packages have been specifically developed to apply atmospheric correction to hyperspectral remote-sensing images, including Atmospheric Correction Now (ACORN) and the Fast Line-of-sight Atmospheric Analysis of Spectral Hypercubes (FLAASH) module for Environment for Visualization (ENVI).

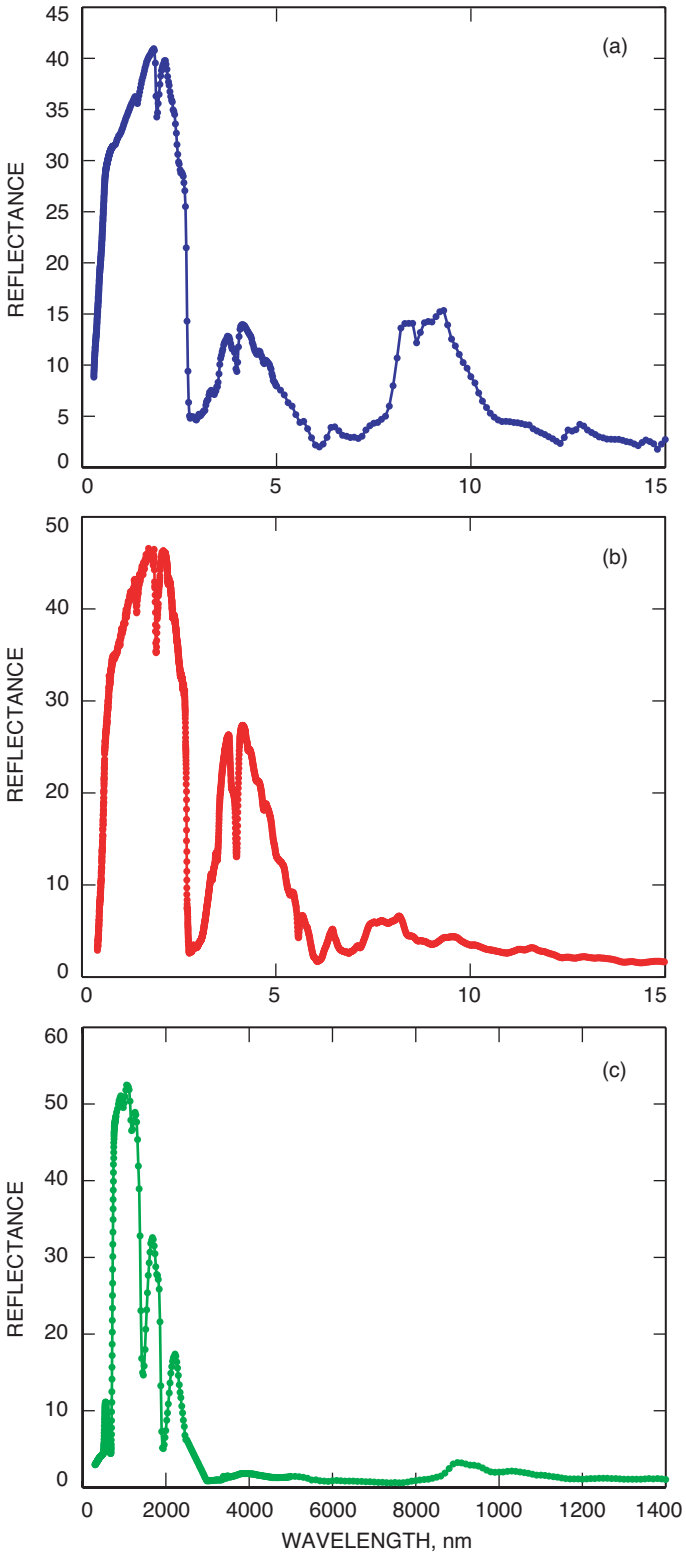

Fig. 3. Sample laboratory spectra, from the ASTER spectral library, with reflectance plotted as a function of wavelength: (a) concrete, (b) soil, and (c) grass.

While it was beyond the scope of this investigation to apply one of these full atmospheric correction models, we attempted to at least partially correct for the effects of the atmosphere, while making some simplifying assumptions. First, we assumed that the atmosphere was constant over each of our scenes. Second, we assumed that the effect of the atmosphere on the measured spectra was entirely multiplicative, with a (possibly different) scaling factor for each band. Finally, we assumed that we could approximately derive the atmospheric correction band ratios by comparing measured spectra over identifiable regions to laboratory spectra of the same substances.

Given these assumptions, we identified regions in a sample image that corresponded to each of the spectra shown in Fig. 3. For each region, we empirically calculated the correction vector needed to align the observed spectra with the laboratory spectra, and then computed a single vector as the average of the regional correction vectors, weighted by the size of each region. In the experiments that follow, we applied this atmospheric correction vector via multiplication to the raw data prior to analysis.

B. Classification and Clustering

If the set of all possible materials is known ahead of time, a *classification* algorithm can be used to label each pixel with its best-match material (e.g., see [6]). However, this approach does not allow for novel materials to be present and discovered. For example, if the spectral library included only {"soil," "grass," "concrete"}, and the image contained pixels where ice had formed, the ice would have to be assigned to one of the three spectral library materials. It is likely that it would be classified as "concrete," due to its brightness, and the fact that something unusual (ice) was present would never be noted. Methods that assign a confidence to each item's classification, and then group all low-confidence items into an "unclassified" set, can alleviate this problem: the ice should receive a very low confidence when classified as "concrete." However, this approach is less effective when multiple unknown classes are present, because they will not be distinguished.

A key strength of our approach is that the algorithm is not restricted to the set of specified materials. Unlike a classification algorithm, which must assign each item (pixel) to one of a finite set of possible classes, clustering methods seek to define data-dependent classes (clusters) that fit the actual observations. This clustering algorithm combines the strengths of both paradigms; it can leverage existing knowledge about material classes while simultaneously discovering new classes, as illustrated in Fig. 2. That is, the seeded k-means algorithm can identify additional, unexpected materials (when $|S| < k$). In addition, it is not forced to find all of the materials in S; if a particular material in S is not present in the image data, then that cluster will be ignored and 0 items will be assigned to it.

IV. Experimental Results

To evaluate the utility of the proposed algorithm, we used it to summarize two very different hyperspectral images. After describing the data sets and experimental procedure, we present results that assess the efficiency of our method and the quality of the resulting summaries. We find that incorporating spectral library information via seeded clustering improves both runtime (by up to 40 percent) and scientific quality (in terms of match to known materials).

A. Experimental Setup and Data Description

Hyperion is a hyperspectral imager that is currently operating onboard the EO-1 satellite. It collects images at 220 wavelengths, from 0.4 to 2.5 μm. Each image is 256 pixels wide, but the length of the image is different for each observational target. We will examine two particular Hyperion images that contain very different terrain: "Arizona" is a desert scene from Arizona; see Fig. 4(a); while "MtEtna" contains city, vegetation, clouds, water, soil, and lava in the area around the volcano, Mt. Etna, in Sicily, Italy. In each case, we ran k-means with and without the spectral library information across each scene 10 times. Each trial used a different random seed, which determines the pixels selected to seed clusters

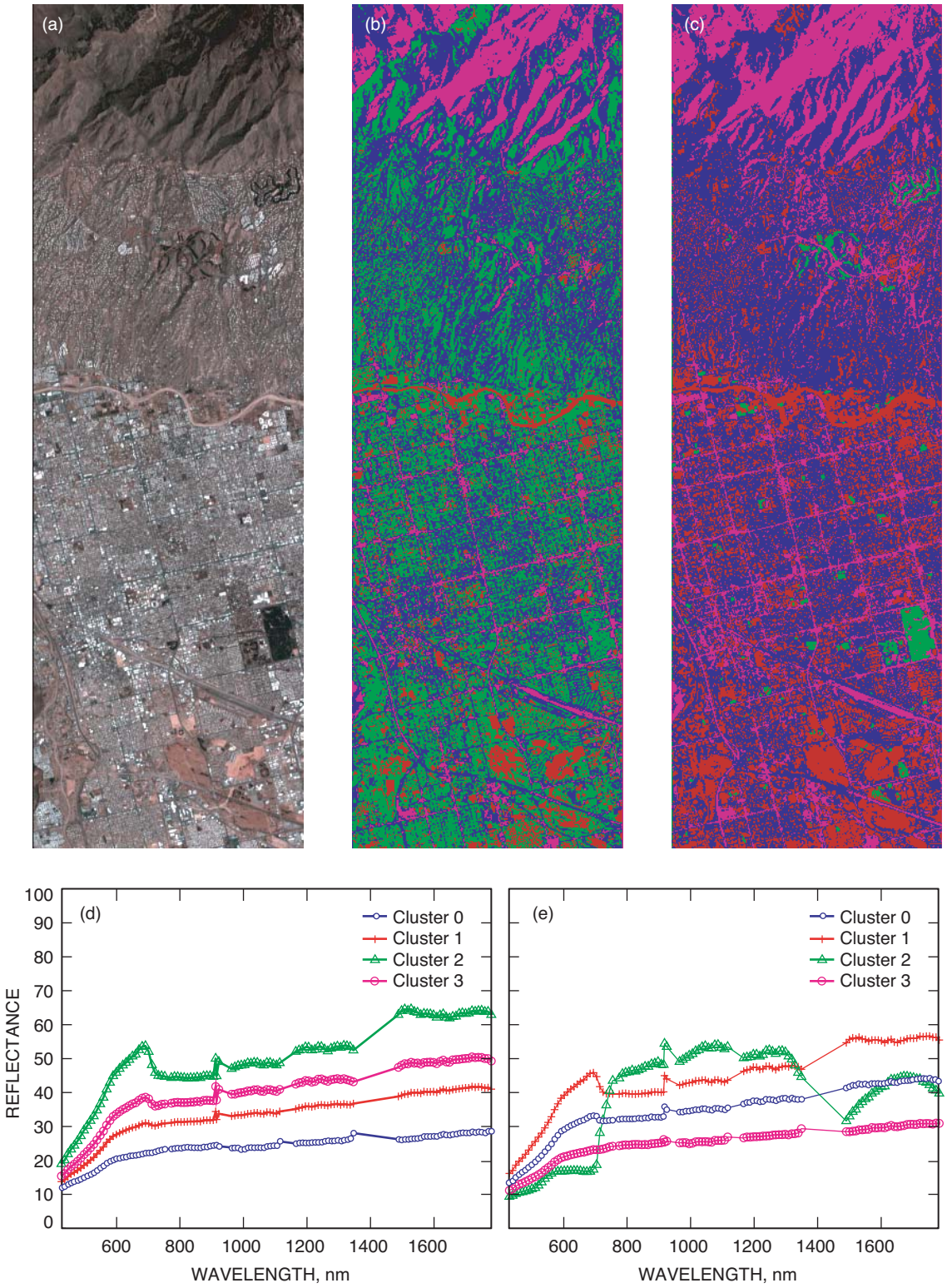

Fig. 4. Cluster means obtained when clustering "Arizona" without, and with, the spectral library present: (a) RGB image, (b) pixel classification map obtained by regular *k*-means clustering, (c) pixel map obtained by clustering with the spectral library, (d) regular clustering: cluster means, and (e) with spectral library: cluster means.

that do not have spectral library information associated with them. Where specific results are presented, we show the most common result that was obtained.

B. Hyperion Hyperspectral Image Summaries

We analyzed "Arizona" both with and without the spectral library present, for $k = 4$ clusters. When using the spectral library, we specified three cluster seeds (concrete, soil, and grass) and left the fourth cluster unseeded. This approach enables a combination of guided (seeded) and exploratory (unseeded) analyses of the data set. As previously discussed, a thorough analysis of this data set would involve clustering with several values for k, permitting the identification of finer divisions inside the discovered clusters. However, a single run with $k = 4$ suffices to demonstrate our approach.

The pixel maps shown in Figs. 4(b) and 4(c) show the output per-pixel assignment to clusters, as indicated by the color of each pixel. We see clearly that, when the spectral library is used, the green cluster stands out as a spatially coherent material. Higher resolution airborne imagery of the same region indicates that these pixels correspond directly to golf courses. Seeding the cluster with the "grass" spectrum was very effective in identifying grassy regions, which were not discovered by the regular, unseeded, k-means analysis.

Figures 4(d) and 4(e) show the cluster centers obtained for both clustering methods. Regular clustering, in Fig. 4(d), produces clusters that are distinguished from each other mainly in terms of intensity (brightness). This is a common result when clustering hyperspectral data. Since the clusters differ in brightness rather than characteristic spectral features, no compositional inferences are possible. In contrast, Fig. 4(e) shows that the clusters obtained when using the spectral library tend to differ in meaningful ways (particularly cluster 2, the "grass" cluster).

Figure 5 shows each seeded cluster as compared to the library spectrum that was used to seed it. Note that the x-axis shows a more limited wavelength range than the plots in Fig. 3; this is because the spectral library elements were pruned to include only wavelengths that matched with Hyperion's observed wavelengths. We see that the seeded and converged cluster means are very good matches. Quantitatively, we calculate the degree of mismatch between the seed and the converged center as their mean squared distance, in units of reflectance (lower values indicate a better match):

Material	Concrete	Soil	Grass
Distance	3.49	10.72	8.98

We performed a similar analysis of the "MtEtna" data set with k set to 4 clusters, this time seeding three of the clusters with {"water," "soil," and "grass"} materials, and leaving the fourth unspecified. The match to the spectral library is not as close as for the "Arizona" data set, but is still reasonable:

Material	Water	Soil	Grass
Distance	10.60	10.98	25.57

In this case, the match to "grass" is worse because this image contains several clouds that were clustered together with the vegetation. A higher value for k would allow us to separate these out, and in fact, the high mismatch value is a useful signal, indicating that the cluster was forced to deviate significantly from its seed.

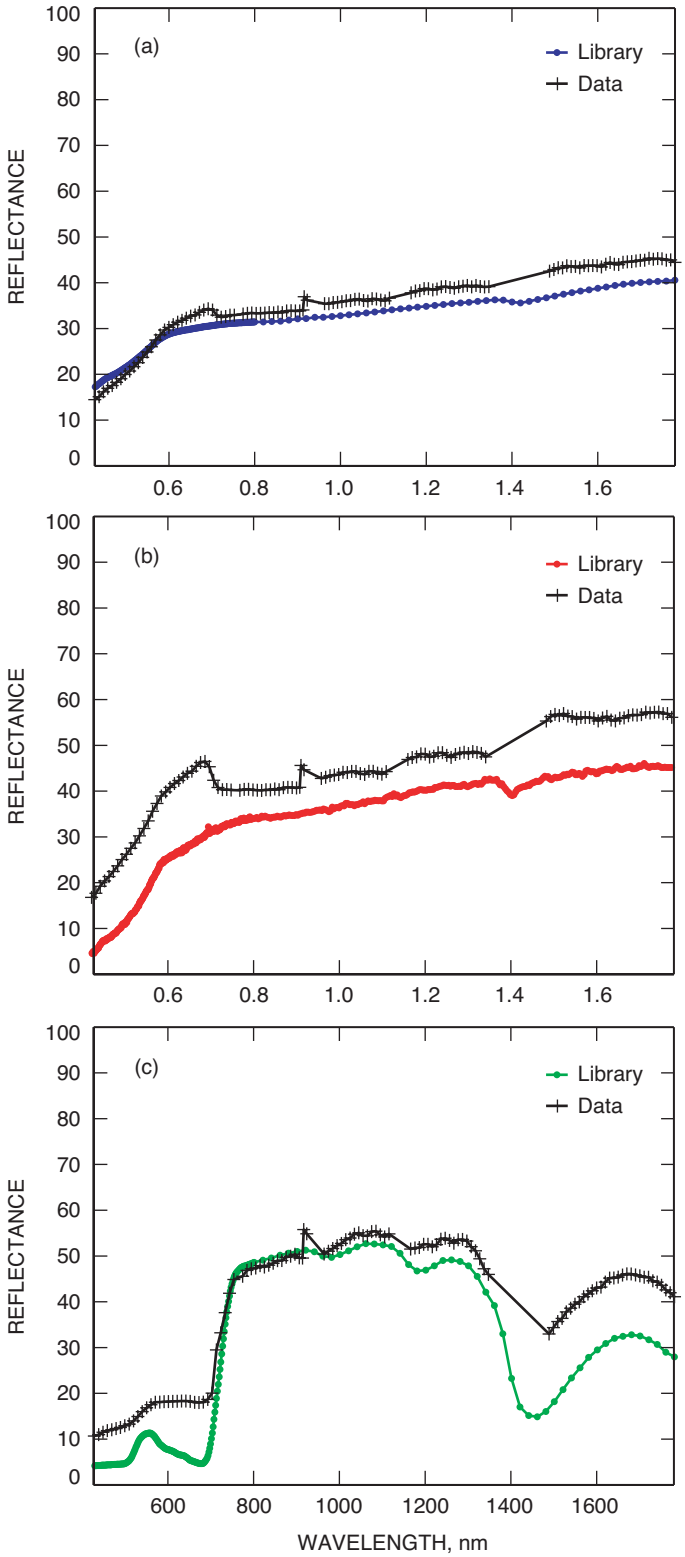

Fig. 5. Converged cluster means, with their cluster seeds included for comparison, for "Arizona": (a) cluster 0, concrete, (b) cluster 1, soil, and (c) cluster 2, grass.

C. Runtime Benefits

In addition to providing more interpretable summaries, we find that clustering with information from the spectral library also results in reduced runtime. Because the clustering algorithm has "hints" about what the clusters should look like, it does not need to spend as much time searching for a good solution. When analyzing "Arizona" with $k = 4$, we find that the mean runtime (evaluated over 10 trials) drops from 465 seconds without the spectral library to 275 seconds with it, a runtime reduction of 40 percent. In addition, the runtime behavior is more consistent; the standard deviation in runtime drops from 204 to 49 seconds when the spectral library is used. Similarly, with the "MtEtna" data set, we find that the mean runtime drops from 1500 seconds to 899 seconds (also a 40 percent reduction), and the standard deviation drops from 827 to 278 seconds when the spectral library is used.

V. Conclusions

In this article, we have presented a method for clustering, or summarizing, large hyperspectral data sets while taking advantage of existing information in the form of spectral libraries. A scientist using clustering methods to analyze spectral data can specify what materials are likely to be present in the image before the analysis begins. These cluster "seeds" can be specified for some, or all, of the clusters. In addition, the algorithm does not require that each material specified be present; if insufficient evidence for a given material is present in the data, that cluster will end up empty. Only materials with reasonable support from the observed data will persist to the final clustering summary. This approach is more efficient than regular clustering and results in image summaries that are more informed and interpretable.

In future work, we wish to investigate the possibility of specifying a strength for each cluster seed. This option would allow users to indicate their confidence in the likelihood of each cluster seed actually being present in the data set.

Acknowledgments

We would like to thank Ben Cichy and Nghia Tang for their assistance in working with Hyperion data. We would also like to thank the entire OASIS (Onboard Autonomous Science Investigation System) team for their support. This work was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration. It was funded by the Interplanetary Network Directorate and NASA's Intelligent Systems Program. The spectral library data presented here were reproduced from the ASTER Spectral Library through the courtesy of the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California.

References

[1] S. Basu, A. Bannerjee, and R. Mooney, "Semi-supervised Clustering by Seeding," *Proceedings of the Nineteenth International Conference on Machine Learning*, Sydney, Australia, pp. 19–26, 2002.

[2] A. Berk, G. P. Anderson, L. S. Bernstein, P. K. Acharya, H. Dothe, M. W. Matthew, S. M. Adler-Golden, J. H. Chetwynd, Jr., S. C. Richtsmeier, B. Pukall, C. L. Allred, L. S. Jeong, and M. L. Hoke, "MODTRAN4 Radiative Transfer Modeling for Atmospheric Correction," *Proceedings of the 1999 Meeting of SPIE (Society of Photo-Optical Instrumentation Engineers)*, vol. 3756, p. 348–353, 1999.

[3] M. Bilenko, S. Basu, and R. J. Mooney, "Integrating Constraints and Metric Learning in Semi-supervised Clustering," *Proceedings of the Twenty-first International Conference on Machine Learning*, Banff, Canada, pp. 11–18, 2004.

[4] P. S. Bradley and U. M. Fayyad, "Refining Initial Points for k-Means Clustering," *Proceedings of the Fifteenth International Conference on Machine Learning*, Morgan Kaufmann: San Francisco, California, pp. 91–99, 1998.

[5] I. Davidson and S. S. Ravi, "Clustering with Constraints: Feasibility Issues and the k-Means Algorithm," *Proceedings of the 2005 SIAM International Conference on Data Mining*, Newport Beach, California, 2005.

[6] A. G. Davies, S. Chien, V. Baker, T. Doggett, J. Dohm, R. Greeley, F. Ip, R. Castaño, B. Cichy, R. Lee, G. Rabidear, D. Tran, and R. Sherwood, "Monitoring Active Volcanism with the Autonomous Sciencecraft Experiment (ASE) on EO-1," *Remote Sensing of Environment*, in press, 2005.

[7] D. Klein, S. D. Kamvar, and C. D. Manning, "From Instance-Level Constraints to Space-Level Constraints: Making the Most of Prior Knowledge in Data Clustering," *Proceedings of the Nineteenth International Conference on Machine Learning*, Sydney, Australia, pp. 307–313, 2002.

[8] J. B. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," *Proceedings of the Fifth Symposium on Math, Statistics, and Probability*, University of California Press: Berkeley, California, pp. 281–297, 1967.

[9] K. Wagstaff, *Intelligent Clustering with Instance-Level Constraints*, doctoral dissertation, Cornell University, Ithaca, New York, 2002.

[10] K. Wagstaff, "Clustering with Missing Values: No Imputation Required," *Classification, Clustering, and Data Mining Applications (Proceedings of the Meeting of the International Federation of Classification Societies)*, Chicago, Illinois: Springer, pp. 649–658, 2004.

[11] K. Wagstaff and J. F. Bell, "Automated Analysis of Mars Multispectral Observations," abstract, *The Sixth International Conference on Mars*, http://www.lpi.usra.edu/meetings/sixthmars2003/pdf/3120.pdf

[12] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl, "Constrained k-Means Clustering with Background Knowledge," *Proceedings of the Eighteenth International Conference on Machine Learning*, Williamstown, Massachusetts: Morgan Kaufmann, pp. 577–584, 2001.