CHAPTER 6
SOFT CONSTRAINTS AND APPLICATION 3: SPECTRAL ANALYSIS

This chapter presents a real-world[1] problem involving spectral analysis of telescopic observations of Mars. We have previously explored the use of clustering with hard constraints. This task is exploratory in nature, and because our knowledge of Mars is itself approximate and incomplete, encoding that knowledge using *soft* constraints is a necessity. The first major contribution presented in this chapter is the development of a soft constrained clustering algorithm. Our analysis of real Mars data produces the second major contribution: the preliminary conclusions we can draw based on constrained clustering of this data simultaneously confirm previous discoveries and suggest novel hypotheses about the planet. Our techniques are demonstrated to be useful for scientific discovery of new knowledge.

After describing the problem of spectral analysis (Section 6.1), we survey other approaches that use clustering on this problem (Section 6.2). We analyze the performance of the basic k-means clustering algorithm in Section 6.3. In Section 6.4, we present a soft constrained k-means clustering algorithm, and in Section 6.5 we discuss the soft constraints we used in our experiments. We conclude with a discussion of our experimental results in Section 6.6.

6.1 Spectral Analysis

Spectral analysis, or *spectroscopy*, attempts to infer the composition or mineralogy of an object based on measurements of how it interacts with light (electromagnetic radiation) (Hunt, 1980; Bell, 1997). In this case, we focus on reflectance spectroscopy, in which we analyze the difference between the light incident on the object and the light that it reflects. The ratio of reflected to incident light, after being corrected as much as possible for viewing geometry and other experimental factors, is referred to as *radiance* or *albedo*. Due to the arrangement of atoms and their electrons, different elements and compounds produce reflectance spectra with different features. These features include *absorption bands*, which are contiguous regions in the spectra with relatively low radiance, indicating that the object is absorbing incident light strongly at those wavelengths. Other interesting features include the slope of the radiance between specific wavelengths and the location of radiance peaks and valleys.

Once radiance observations have been collected, it is possible to compare an observed spectrum to that of several different minerals or other compounds, as measured in the laboratory, to determine the composition of the remote object. Of course, for real observations, a given region on a remotely observed object is unlikely to be composed purely of one element or compound. It is more common to encounter a mixture, and this means that characteristic spectral features may be muted or absent. Nevertheless, it is often still possible to identify major constituents of an object based on the observed spectrum by performing matching to laboratory spectra (Adams and McCord, 1969).

Infrared observations. We do not restrict our observations to the visible part of the electromagnetic spectrum. The window of visible wavelengths is quite narrow (300 – 700

[1] Real *other* world.

nm), and distinguishing characteristics often appear beyond this limited scope, especially in the near-infrared (700 – 5000 nm). For example, an Fe^{3+} absorption band for various minerals occurs between 850 and 950 nm. Therefore, observations in the near-infrared are very important for remote sensing and for identifying the chemical composition of a remote body (Bell, 1997).

Manual spectral analysis. Data collected by remote sensing is most commonly interpreted and analyzed manually by experts in the field. Generally, this is done by creating radiance ratio images and scatterplots and then analyzing the results visually to construct threshold values that will distinguish homogeneous groups from each other. A *radiance ratio image* is created by calculating the ratio of radiance at two specified wavelengths for each pixel and then scaling the resulting values to produce an output greyscale image. *Scatterplots* are created by plotting pixels in two dimensions, usually either the radiance values at two wavelengths or a radiance value versus a related ratio value. Often, these values will be binned and colored according to the number of pixels that fall into each bin, so that highly-populated areas will be more immediately apparent.

Both of these analysis tools have been instrumental in the discovery of new knowledge about Mars. McSween Jr. et al. (1999) selected "representative" spectra from rock observations obtained by the Mars Pathfinder Lander and used the results of both techniques to conclude that there are four major spectral classes of rocks present. Bell et al. (2000) performed a detailed manual analysis of the dust and soil found at the Pathfinder landing site and determined that there are two distinct "dust" and five "soil" classes. However, by themselves these tools are only an intermediate step in the analysis process. A human expert is called upon to interpret the ratio images and scatterplots.

Consequently, manual analysis has limited scalability. As data sets get larger and larger, there is a real need for automated methods to assist in reducing the number of individual items that a human must examine. Some researchers have already experimented with using techniques from statistics and machine learning to augment their manual analyses of astronomical data. As reported by Lillesand and Kiefer (1987), this work has involved both supervised and unsupervised approaches.

Supervised learning for spectral analysis. The most popular supervised learning technique for spectral analysis is the neural network, although decision trees have also been used (Bertin, 2000). Gulati et al. (1995) applied a neural network to the problem of stellar classification (into known star categories). They selected 55 spectra with known classifications for training and then applied the learned neural network to 158 test spectra. Merényi et al. (1996) used a neural network to analyze spectroscopic observations of Mars; they manually identified ten classes in the data, then labeled five to ten sample training spectra for each class. After training the neural network on these examples, they were able to apply it to the full data set and classify each pixel into one of the ten classes.

Supervised learning can be of help in processing large data sets because human intervention is needed only for the initial labeling of the training set. However, supervised learning does not completely solve the problem of scalability. Gulati et al. note that their approach would be substantially improved by having access to a larger training set, and ideally, one that is more consistent with the items in the test set. It is not easy in general to estimate the number (or identity) of items required for the training set to guarantee good performance on unseen items. More importantly, supervised methods are useful when the goal is to apply an existing categorization to new data. We are in a different situation; we do not know precisely what categorization of Mars observations will yield the most information.

To summarize, the motivating hypothesis behind this work is that an analysis of spectral observations can identify geologically or compositionally interesting regions on Mars. This task is by definition an exploratory one. The volume of data being collected makes manual analysis impractical. Therefore, we turn to automated analysis methods as our tool of choice. Supervised methods would require us to specify examples of what we want to find in the data, while unsupervised clustering algorithms instead adapt to patterns actually present in the data set and therefore are less "biased." In addition, clustering algorithms can rapidly process large data sets and produce summary output (clusters) that point the human analyst towards specific subsets of data that have interesting characteristics. In the next section, we will discuss previous clustering approaches to the problem of spectral analysis.

6.2 Other Clustering Approaches to Spectral Analysis

We begin with a review of various clustering analysis techniques that have been applied to spectral data. Next, we discuss methods that can incorporate additional information. These methods have largely been limited to working with image contiguity information (spatial relationships between pixels).

6.2.1 Clustering Spectral Information

Clustering has been demonstrated to be of use on a variety of astronomy problems. It has primarily been used to automatically classify observations into meaningful groups or to perform image segmentation.

Classification. Djorgovski et al. (2000) discuss the utility of clustering algorithms for analyzing large sky surveys. Yoo et al. (1996) used COBWEB/95, an enhanced version of COBWEB (Fisher, 1987) that can handle continuous attributes, to induce a clustering hierarchy on 33,021 sky objects from the Second Palomar Observatory Sky Survey. The algorithm's taxonomy of sky objects successfully separated stars from galaxies, even though no training data was provided and this particular goal was not specified to the algorithm. de Carvalho et al. (1995) applied Autoclass, a Bayesian clustering algorithm (Cheeseman et al., 1988), to the same data set and obtained a partition of the objects into four clusters. Manual post-analysis of the clusters determined that they represented meaningful categories: stars, galaxies with bright central cores, galaxies without bright cores, and stars with "fuzz" around them.

Image segmentation. For a data set that includes observations taken at d spectral bands, we can view each pixel as a data vector, containing d values. The distance between two pixels can be defined as the Euclidean distance between their respective d-valued spectra. We can then apply common clustering algorithms to organize the data set into subregions that are internally similar and distinct from each other.

A different approach is to cluster the pixels in an image based on a histogram rather than the individual points themselves (Narendra and Goldberg, 1977; Schowengerdt, 1983; Wharton, 1983). Their algorithm effectively discretizes the feature space into a set of cells, where each cell is annotated with a *frequency*, i.e., the number of data points that fall into that region. Clusters are identified by seeking cells with relatively high frequency counts. This approach is computationally efficient and does not need to make any assumptions about the probability distributions of the clusters (e.g., Gaussians). However, these benefits come at the expense of losing fine discriminability, and as with any discretizing operation, the quality of the results is dependent on the choice of number of cells and their placement in feature space.

6.2.2 Incorporating Spatial Information

Although spectral information is very important for remote sensing analysis, it is also desirable to include other relevant information that is available. By far, the largest body of work on spectral analysis that uses information beyond just the spectral features is that which attempts to incorporate the spatial information inherent in a two-dimensional image. Each pixel is described by observations at d spectral bands, but it is a common hypothesis that the relative position of the pixels is also important for classification purposes. In other words, the identity of neighboring pixels may also be important to the classification of a given pixel. This bias is supported by the fact that, for real remote sensing imaging systems, several experimental factors (e.g., diffraction and internal scattering) cause light from a given location in the scene to spread out into neighboring pixels.

K-means variants. One approach is to use a Bayesian update rule inside the iterative k-means algorithm, and to weight the priors so that a given pixel has a predisposition for receiving the same classification as its neighbors (Montolio et al., 1992). Alternatively, Theiler and Gisler (1997) modified the objective function used by k-means to take into account the overall contiguity of the partition. We will adopt this approach for our own development of a soft constrained clustering method, but we will allow the encoding of domain knowledge from a variety of sources, not just spatial contiguity.

Spatial information as additional features. Soh and Tsatsoulis (1999) used the regular COBWEB clustering algorithm to perform image segmentation of images taken at a single wavelength. To do this, they first performed an initial segmentation of the image into regions by automatically finding threshold values such that all pixels within a region have intensity values below the calculated threshold. They then calculated additional spatial and textural features for each such region by examining, for example, the region's spatial *variety* (synonymous with discontiguity). Next, they used the COBWEB clustering algorithm to condense these regions into large-scale clusters in the image.

The authors applied this technique to several Earth images, and the approach demonstrates apparently good results visually. However, they provide limited assessment, despite the fact that they report class labels for each of the clusters obtained from COBWEB.

Quantitative assessment is limited to a comparison of the size of the resulting clusters to the amount of coverage in each real class indicated by a human expert on the same images; we do not know how much these clusters agree (spatially or spectrally) with the expert classification. In addition, this approach has only been applied to images taken at a single wavelength. For multi-spectral data, the authors speculate that their technique could be applied to each wavelength separately, and the results could be fused together in an unspecified manner. In contrast, we will accept multi-spectral input directly into a single run of the algorithm and automatically produce unified output.

Neighbors encoded as additional features. Masson and Pieczynski (1993) developed a stochastic version of the EM clustering algorithm (Dempster et al., 1977) by constructing additional features for each pixel that incorporated a single neighbor's values. Roberts et al. (1996) compared clustering multi-spectral data on the basis of the spectral information alone to clustering the same data with additional spatial information. The original data set, a Landsat image, contains information from seven spectral bands. Spatial information was incorporated by augmenting each pixel with a set of seven additional features for each of its eight neighbors. The biggest disadvantage of this method is the dramatic increase in dimensionality of the data set; in this case, dimensionality rises from $d = 7$ to $d = 63$. The authors compensated by running a principal components analysis (PCA) to provide a projection of the data down into a two-dimensional space. The PCA and projection were done separately on the original data ($d = 7$) and on the spatio-spectral data ($d = 63$). This transformation of the data allowed the authors to apply a variety of clustering algorithms (to each set of two-dimensional data) without modifying the algorithms themselves. Evaluations of the contiguity of the resulting partitions show that the contiguity is higher when using the spatio-spectral data set.

One of the algorithms Roberts et al. used was the contig-k-means algorithm developed by Theiler and Gisler (1997), which incorporates contiguity information directly into the k-means algorithm as previously described. Thus, when this algorithm was applied to the spatio-spectral data, it was in a sense receiving the same information twice (in the data features and in the spatial relation, which it was able to directly observe). Not surprisingly, the difference in its performance on the two data sets (in terms of contiguity) was very small. However, the variance of the partition it produced on the spatial data was much higher than that of its partition of the spatio-spectral data, which indicates that the latter clusters were a "tighter" fit to the data. On the other hand, it is unclear whether these numbers can be directly compared, since each data set is using a different pair of (PCA-reduced) features to represent each pixel.

6.2.3 The Need for Constraints in Remote Sensing

Although automated tools are useful for summarizing large quantities of data, there is a need for such tools that can also incorporate additional domain knowledge into their analyses. As we have just shown, most previous work with specializing clustering algorithms for the problem of spectral analysis has focused solely on methods for incorporating spatial contiguity. We claim that there is a much richer class of relevant information that can, and should, be used when performing a clustering analysis of spectral data. For example, Pieters

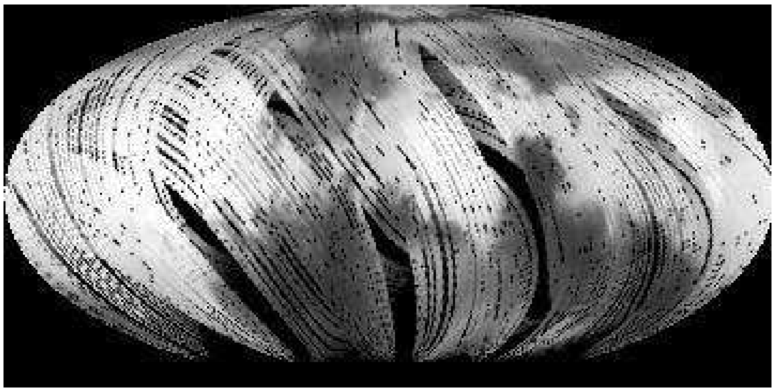


Figure 6.1: Mars observed by STIS at 907 nm

et al. (1996) warn of the dangers when blindly attempting to find the best numerical fit to data. Unsupervised application of their mixture modeling approach to determining mineralogical composition of remote objects often came up with physically implausible results. Therefore, they recommend the use of "additional constraints or assumptions" to restrict solutions "to be consistent with laboratory trends." Further, Djorgovski et al. (2000) warn that "a blind application of the commonly used clustering algorithms in such real-life cases could produce some seriously misleading or simply wrong results."

Exploratory analysis should be guided and constrained as much as possible, to ensure that the results it produces are reasonable from a scientific view. Therefore, our goal is to develop a method that can effectively encode and apply domain knowledge to improve its analysis of spectral data.

6.3 K-means Performance on Spectral Analysis

In the next section, we will show how the basic k-means algorithm can also be transformed to handle soft constraints and argue that this is a powerful and natural way to encompass domain knowledge from a variety of sources. However, we first describe the data set used in our experiments. In addition, we will discuss the performance of the regular k-means algorithm on this problem and show examples of its output.

6.3.1 Experimental Methodology

The data set under consideration is composed of multiple observations of the planet Mars by the Hubble Space Telescope (HST) in April and May, 1999. We will refer to it as the STIS[2] data set. The planet was observed at 1024 different wavelengths, resulting in 1024 images of Mars, each at a resolution of about 20×80 km per pixel. The images at each wavelength were then map-projected onto a 1200×600 pixel latitude/longitude grid to preserve this spatial resolution. To simplify the presentation of our analysis, we will show the results of working with a smaller version of this data set, which was subsampled to

[2]STIS stands for "Space Telescope Imaging Spectrograph," the HST instrument used to collect the data.

Table 6.1: Derived features for the STIS data set

Feature	Description	Calculation	Min	Max
FL	"flatness" of spectrum	$\frac{R_{528}}{R_{762}}$	0.38	1.02
BD_{670}	670 nm band depth (Fe^{3+})	$1 - \frac{R_{667}}{.58*R_{608}+.42*R_{747}}$	-0.061	0.24
BD_{860}	860 nm band depth (Fe^{3+})	$1 - \frac{R_{879}}{.68*R_{801}+.32*R_{1016}}$	-0.02	0.07
BD_{950}	950 nm band depth (Fe^{2+})	$1 - \frac{R_{947}}{.26*R_{747}+.74*R_{1016}}$	-0.07	0.06
SL	800 – 1000 nm slope	$R_{1016} - R_{801}$	-0.05	0.01

360×180 pixels. We also reduced the number of distinct wavelengths under consideration to 26, by performing a sliding Gaussian average on each full spectrum and recording every 40th value. Our final data set, then, contains 64,800 pixels, with a dimensionality of 26. The wavelengths span 528 to 1016 nm.

Figure 6.1 shows the entire planet when observed at a single wavelength (907 nm). The image is a Mollweide equal-area projection with the north pole at the top, south pole at the bottom, and 0^o longitude in the center of the map. Brighter areas indicate a higher observed radiance; this image has been contrast-enhanced to more clearly delineate the light and dark regions. Our data set is an image cube composed of 26 such images.

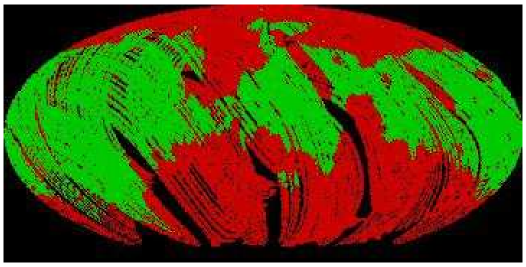
Derived features. In addition to the 26 wavelengths available in the downsampled data set, we computed five additional *derived* features. This choice of additional features was motivated by the features reported as most useful for classifying scenes obtained by the Mars Pathfinder Lander (Bell et al., 2000). The list of derived features is given in Table 6.1. R_λ refers to the radiance observed at wavelength λ.

The first feature of interest is a measure of how flat each pixel's spectrum is (FL), which is calculated by taking the ratio of the radiance at the lowest available wavelength to the radiance at the average "peak" of the STIS spectra. Most of Mars is dominated by areas with very strong absorption at lower wavelengths, which is what gives the planet its characteristic reddish color. The very red regions will have an FL value well below 1.0, while other areas will be distinguished by a FL value near 1.0. For this data set, FL varies from 0.38 to 1.02.

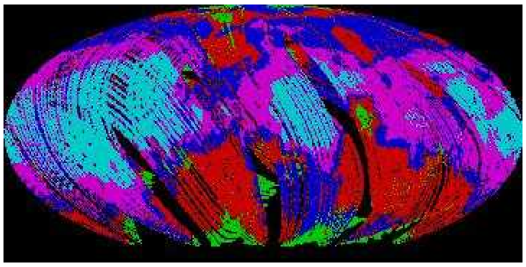
Since we are interested in determining the chemical composition of areas on Mars, we also include features that are sensitive to specific absorption bands. A band centered around 950 nm indicates the presence of Fe^{2+}, while bands near 670 and 860 nm indicate the presence of Fe^{3+} (see Adams, 1974; Morris et al., 1985). (The precise wavelength values differ slightly because we are working with a discrete set of 26 wavelengths, and in each case we selected the closest available value.) The band depth at wavelength λ, BD_λ, is calculated as:

$$BD_\lambda = 1 - ER_\lambda \tag{6.1}$$

where ER_λ is the "expected" radiance at λ in the absence of the band. We determine the "expected" radiance by calculating the continuum value: we compute a linear interpolation between the radiance values at two nearby wavelengths outside of the band.



(a) $k = 2$



(b) $k = 5$

Figure 6.2: Two classifications of Mars by the k-means algorithm

Finally, we also know that the spectral slope between 800 and 1000 nm is important for distinguishing Martian spectra (see McSween Jr. et al., 1999; Bell et al., 2000). We calculate this value by subtracting the radiance at 801 nm from the radiance at 1016 nm. Technically, this slope (SL) should be divided by the wavelength separation (215 nm), but since this value is constant for all spectra, we omitted that additional step.

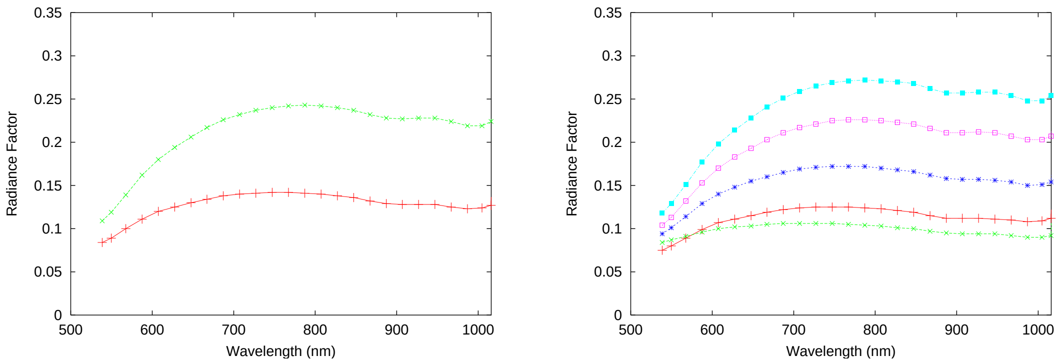
6.3.2 K-means Results

To provide a basis for later comparisons, we ran the regular k-means algorithm on the STIS data set. The output provided by this algorithm indicates what can be learned from an automated analysis of the data set without any specific domain knowledge. The input data includes the 26 spectral observations and five derived features for each pixel.

Before describing the results we obtained with k-means, we will first justify our choice of k-means for this problem. We do not claim that k-means is the only useful clustering algorithm for this problem. It may also be instructive to analyze the data using, for example, an algorithm which attempts to find a good value for k on its own, rather than having it specified as input. However, k-means is useful as an exploratory tool for spectral analysis precisely because we can control the number of clusters it generates. This allows us to examine Mars at different levels of generality. As we will show, the output at different k values provides us with different insights about the planet.

Figure 6.2 shows the output of the k-means algorithm for $k = 2$ and $k = 5$. In these images, each pixel is colored according to its classification by the clustering algorithm (the color values themselves are not meaningful). Black regions correspond to non-planet regions or to missing data. For $k = 2$, the planet is divided into two regions that correspond directly to its characteristic light and dark areas (cf. Figure 6.1). With larger values of k, such as $k = 5$, the planet splits into finer distinctions.

An examination of the average cluster spectra reveals that the cluster separations are dominated by simple albedo differences. Figure 6.3 plots, for each cluster identified by k-means, the average spectrum for that cluster. The x-axis is wavelength in nm, and the y-axis is the radiance factor, I/F. This value is defined as the observed radiance (I) divided by the radiance of sunlight reflected off a "perfect" diffuse reflector at the same distance as Mars (F). The colors used for each spectrum correspond to the colors used in Figure 6.2. Visually,



(a) $k = 2$

(b) $k = 5$

Figure 6.3: Average cluster spectra for the partitions shown in Figure 6.2

the average spectrum in each cluster is very similar to the other cluster spectra in shape, and brightness is the major distinguishing factor. This is not surprising; a principal components analysis (PCA) of this data set (and other Mars observations in general) confirms that the first principal component is albedo and that this component typically explains 90–95% of the variance in spectral observations of Mars (Bell, 1992). An exception to this general rule is the cluster identified by green x's in Figure 6.3b; its shape differs in important ways from the rest of the average cluster spectra. We will return to this interesting exception in Section 6.6.

6.4 Constrained Clustering With Soft Constraints

We have noted in previous chapters that some kinds of domain knowledge are perhaps better encoded as soft constraints (or *preferences*) rather than as hard constraints. This is particularly true for our spectral analysis of the Mars data. We do not yet know precisely what can be found in this data set, but we have some general knowledge about the problem domain that should be of use to a clustering algorithm. Therefore, before proceeding to describe the constraints that we used for this problem, we first describe our formulation for soft constraints and present an algorithm that can use them, SCOP-KMEANS.

6.4.1 Soft Constraints

Previously, we defined two kinds of hard constraints, must-link and cannot-link (see Section 3.1.1). Each constraint was specified as a relationship between two data items. For soft constraints (preferences), we will augment each relationship with an additional *strength* factor, s, that indicates how reliable the constraint is. This single constraint formulation ($\langle d_i, d_j, s \rangle$) subsumes both soft and hard constraints. The absolute value of s ranges from 0 to 1, with higher values indicating a stronger constraint. We also attach a sign to the strength; negative values indicate a preference against being grouped together, while positive values indicate a preference towards being grouped together. A constraint such as $\langle d_i, d_j, 1 \rangle$ is equivalent to a must-link constraint, and $\langle d_i, d_j, -1\rangle$ is equivalent to a cannot-link constraint ($\langle d_i, d_j, 0\rangle$ is a "don't-care" statement).

Soft constraint closure. The use of soft constraints requires a modification to how we compute the constraint closure. The graph of the constraint relation is weighted; each edge (constraint) has a weight determined by its strength. In Section 3.1.1, we defined the closure of a set of hard constraints as follows:

$\forall i, j, k$: given		produce
$d_i =_m d_j$	$d_j =_m d_k$	$d_i =_m d_k$
$d_i =_m d_j$	$d_j \neq_c d_k$	$d_i \neq_c d_k$
$d_i \neq_c d_j$	$d_j =_m d_k$	$d_i \neq_c d_k$

Because the edges are now weighted, each derived constraint must also have a strength assigned to it. We calculate this *conservatively* as the minimum (absolute) strength of the two constraints that imply the new constraint. The closure for a set of soft constraints is defined in this way:

$\forall i, j, k$: given		produce
$\langle d_i, d_j, s_1\rangle$	$\langle d_j, d_k, s_2\rangle$	$\langle d_i, d_k, min(s_1, s_2)\rangle$
$\langle d_i, d_j, s_1\rangle$	$\langle d_j, d_k, -s_2\rangle$	$\langle d_i, d_k, -min(s_1, s_2)\rangle$
$\langle d_i, d_j, -s_1\rangle$	$\langle d_j, d_k, s_2\rangle$	$\langle d_i, d_k, -min(s_1, s_2)\rangle$

where s_1 and s_2 are the (positive) strength values for the corresponding constraints. If both constraints are positive (preference towards linking), then the derived constraint also has a positive strength, which is the minimum of s_1 and s_2. If one of the contributing constraints is negative, then the derived constraint also has a negative strength, which is the minimum of s_1 and s_2. As before, if both contributing constraints are negative, then we cannot conclude anything directly from them, so we do not create a derived constraint.

6.4.2 The SCOP-KMEANS Algorithm

We have chosen k-means as our prototype for the development of a soft constraint clustering algorithm. As discussed above, this is a useful algorithm for the spectral analysis problem, so it is very relevant to the goals of this chapter. A similar transformation is possible with the COBWEB clustering algorithm, but we will not present those details explicitly here.

We achieve soft constrained clustering with k-means by modifying its objective function to incorporate a real-valued penalty for violating constraints. This penalty is proportional to the strength of the violated constraint(s). SCOP-KMEANS[3] calculates CV, the maximum strength of the violated constraints, if any. See Table 6.2 for a full listing of the SCOP-KMEANS algorithm; once again the changes from the basic k-means algorithm are shown in bold. The objective function, f, combines CV with var (variance) in the following way:

$$f(C_1 \dots C_k) = \frac{var}{1 - CV} \tag{6.2}$$

[3] "S" stands for "soft."

Table 6.2: SCOP-KMEANS algorithm

SCOP-KMEANS(number of clusters k, data set D, preferences $Pref\ \langle a,b,s\rangle$ where $(a,b) \subseteq D \times D$ and $s \in [-1,1]$)

1. Let $C_1 \ldots C_k$ be the initial cluster centers.
2. For each instance d in D, assign it to the cluster C_j which will minimize the following objective function:
$$f(C_1 \ldots C_k) = \frac{var}{1 - CV}$$
where var is the variance of the partition and CV is the constraint violation value; $CV :=$ CONSTVIOL$(d, C_1 \ldots C_k, Pref)$**.**
3. Update each cluster center C_i by averaging all of the points $d_j \in C_i$ that have been assigned to it.
4. Iterate between (3) and (4) until convergence.
5. Return the partition $\{C_1 \ldots C_k\}$.

CONSTVIOL(data point d, partition $C_1 \ldots C_k$, preferences $Pref$)

1. Let $CV_{max} := 0$, $nConst := 0$, $nViol := 0$.
2. For each $\langle d, d', s\rangle \in Pref$:
   If $|s| > CV_{max}$,
   If $s > 0$ and $d.class \neq d'.class$, then $CV_{max} := |s|$ and $nConst := nViol := 1$.
   Else if $s < 0$ and $d.class = d'.class$, then $CV_{max} := |s|$ and $nConst = nViol := 1$.
   Else if $|s| := CV_{max}$,
   Increment $nConst$ by 1.
   If $s > 0$ and $d.class \neq d'.class$, then increment $nViol$ by 1.
   Else if $s < 0$ and $d.class = d'.class$, then increment $nViol$ by 1.
3. Return $CV_{max} * \frac{nViol}{nConst}$.

---

where CV is an estimation of the proportion of the maximum strength constraints that are violated, weighted by their strength. It ranges from 0 (no constraints violated) to 1 (all constraints with strength 1.0 violated). The penalty for constraint violation is non-linear. Since the goal is to minimize the objective function, partitions that violate constraints of greater strength will appear much "worse" to the algorithm than those that violate weaker constraints.

More precisely, the CONSTVIOL function examines each constraint that applies to data point d. CV_{max} is the maximum (absolute) strength of d's violated constraints. $nConst$ counts the number of constraints at strength CV_{max}, and $nViol$ tracks how many of those are violated. The function then returns CV_{max} times the fraction of maximum strength constraints that are violated. Constraints at a lower strength may or may not also be violated, but the maximum strength violations take precedence, as they should.

This algorithm is similar in motivation to the contig-k-means algorithm (Theiler and Gisler, 1997). However, they incorporate only one kind of soft constraint into the k-means objective function (contiguity), and they calculate a linear combination of contiguity and variance:

$$f(C_1 \ldots C_k) = \lambda \cdot D + (1 - \lambda) \cdot var \tag{6.3}$$

where D refers to the discontiguity of the partition $C_1 \ldots C_k$, and λ is a user-specified weighting factor that determines the relative importance of variance versus contiguity. In contrast, SCOP-KMEANS has the ability to incorporate a variety of different soft constraints, each with their own strength. In particular, our formulation can encode information indicating stronger contiguity in certain areas than in others. Further, our algorithm can accommodate negative constraints, which the contig-k-means algorithm cannot. Their constraint model does not allow negative constraints.

6.5 Generating Spectral Analysis Constraints

SCOP-KMEANS gives us the ability to encode approximate or heuristic domain knowledge in a form that the algorithm can use to guide its search through the space of possible partitions of the data. With respect to our exploratory analysis of HST observations of Mars, soft constraints are essential. Although some knowledge is encoded in our choice of features (and derived features), other information cannot be represented in that form. In contrast to previous approaches to clustering spectral data, we can use constraints to make such information available to the clustering algorithm.

There are two major kinds of domain knowledge that we selected for testing in conjunction with constrained clustering. The first, *spatial contiguity*, is commonly useful whenever processing images. The second, *slope relationships*, is specific to our current knowledge of the planet Mars. Both kinds of information cannot be effectively encoded either as simple features or as hard constraints. Although we will only report on two sources of domain knowledge for this problem, we do not claim that they are a comprehensive representation of everything currently known about Mars. A significant advantage of our clustering architecture is that new hypotheses about data relationships can be easily encoded and tested. We plan to investigate several additional sources of domain knowledge about this problem in a similar way in the future.

Spatial contiguity. The assumption that regions in an image should be spatially contiguous is a common one. When performing image segmentation to identify objects in a scene, contiguity is required.

For remote sensing, contiguity is important, but there is an important tradeoff between contiguity and the level of precise distinctions we wish to make. For example, we could classify the entire state of Utah as "desert," or we could identify multiple regions ("canyons," "desert," "forest," "water"). However, the "water" cluster will appear in several different isolated regions of the state, which greatly reduces the contiguity of this cluster (and the entire partition). Creating a separate cluster for each body of water ("Great Salt Lake," "Utah Lake," "Colorado River," "Lake Powell," etc.) would improve the contiguity of the partition, but this is only useful if the water in each of these bodies is different enough (in the end user's opinion) to merit being placed in different clusters.

Therefore, we express spatial contiguity as a soft constraint. It is a *global constraint*, but we can easily encode it as a set of instance-level soft constraints. More precisely, for each pair of pixels d, d' that are spatial neighbors in the image, we generate a constraint $\langle d, d', s_{contig} \rangle$. Formally,

$$\forall_{d,d'} \; d.neighbor(d') \Rightarrow \langle d, d', s_{contig} \rangle \in Pref.$$

We define the spatial neighbors of d as its eight directly adjacent pixels (border pixels will have fewer than eight neighbors). This formulation allows us to easily experiment with different values for s_{contig}. We will show the results obtained with a range of s_{contig} values in Section 6.6.

Slope relationships. The SL derived feature, which indicates the slope of a given spectrum between 800 and 1000 nm, is known to be very important for distinguishing Mars spectra (Bell et al., 2000). A negative slope can be indicative of a Fe^{2+} absorption band, hinting at the presence of olivine or pyroxene. However, a positive slope rules out the possibility of such a band. Therefore, because we are seeking mineralogically significant separations, we decided to test applying a negative soft constraint to pairs of spectra that have slopes of different signs. In this case, we are encoding a *feature-level constraint* as a set of instance-level constraints. More precisely,

$$\forall_{d,d'} \; d.SL > 0 \; and \; d'.SL < -0.03 \Rightarrow \langle d, d', s_{slope} \rangle \in Pref.$$

The range of SL on this data set is from -0.05 to 0.01. We restrict this constraint to items d' with a relatively strong negative value ($d'.SL < -0.03$). We will show the results of experimenting with various fixed values of s_{slope}. Another possible approach would be to let s_{slope} vary with the magnitude of the difference between $d.SL$ and $d'.SL$. Again, this level of fine control over soft constraints has not previously been possible.

6.6 Experimental Results

Our experiments with the STIS data and soft constraints have yielded a variety of interesting and exciting results. We have organized the presentation of these results into three

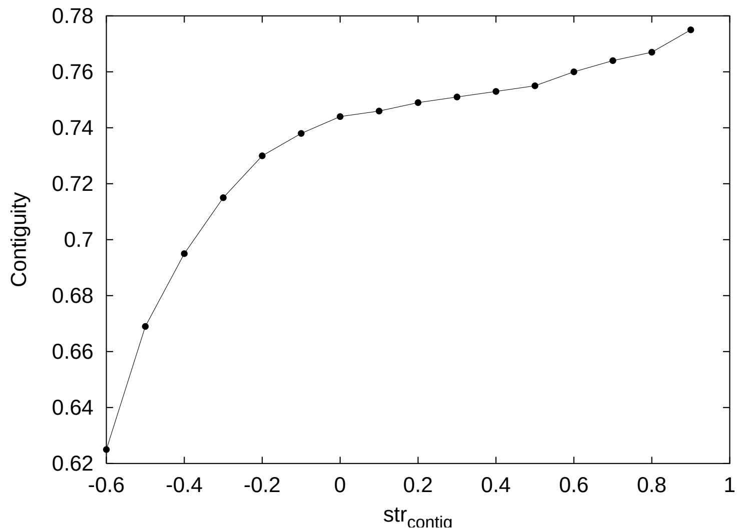


Figure 6.4: Contiguity obtained with k=4 for different s_{contig} values

categories. First, we will show experiments that confirm that our encoding and use of soft constraints is in fact an effective way to incorporate domain knowledge in clustering. Next, we will compare our analysis to results reported by other researchers about Mars. Finally, we will present a summary of several novel results that our experiments have produced. The results of these experiments, taken as a whole, firmly establish constrained clustering as a useful spectral analysis tool.

6.6.1 Effective Use of Soft Constraints

The spatial contiguity constraint incorporates the spatial relationships in an image into the clustering process. We can test whether this information has been effectively encoded and applied by the algorithm by looking at the contiguity of the resulting partition with and without constraints. This kind of constraint therefore provides an easy way to confirm the effectiveness of our soft constraint formulation and algorithm.

With respect to the STIS data, we created a set of contiguity constraints as described in Section 6.5 and experimented with different values for s_{contig}, the constraint strength parameter. In this case, we do not take a transitive closure over these constraints. This heuristic encodes a local notion of contiguity, which indicates that a pixel prefers to be classified similarly to its neighbors. A transitive closure of these constraints would require that *every* pixel be placed into the same class (with strength s_{contig}), which is not the preference we want to encode.

Figure 6.4 plots the per-pixel contiguity of the output partition for sixteen different runs of SCOP-KMEANS on the STIS data with $k = 4$. Each run used the same random initial cluster centers but had access to a different set of soft constraints; each constraint set used a different value for s_{contig}. Contiguity on the y-axis is specified as the average per-pixel contiguity. It is calculated in the following way:

$$Contig(P) = \frac{1}{n} \sum_{(d,d') \in N} \delta(d.class = d'.class). \tag{6.4}$$

The partition P contains all of the items $d \in D$, with each item annotated with its assigned cluster (class). N is a binary relation containing neighbor pairs (d, d'). δ is a function that returns 1 if its argument is true and 0 otherwise. A per-pixel contiguity of 0.75, for example, means that on average, 3/4 of a pixel's neighbors have the same class that the pixel does.

We experimented with s_{contig} ranging from -0.9 to 0.9 (we will explain what a negative value for this constraint means shortly). There are no solutions for $s_{contig} = 1.0$. The only way to find a valid solution for $s_{contig} = 1.0$ is to put all of the pixels into the same class, which is impossible for any $k \neq 1$. Also, the results obtained with $s_{contig} = [-0.7, -0.9]$ are degenerate; clusters are composed of uniformly scattered points in the image.

Figure 6.4 shows that as s_{contig} increases, so does the contiguity of the output partition, which demonstrates that the SCOP-KMEANS algorithm can accept these soft constraints and apply them successfully. Normally, we would use a positive value for s_{contig}, to indicate a preference towards spatial contiguity. However, we also tested what would happen when specifying a *negative* value for s_{contig}, indicating a preference against contiguity. SCOP-KMEANS was also successful at incorporating this kind of constraint: contiguity steadily decreases as s_{contig} becomes more negative.

The output at $s_{contig} = 0.0$ is exactly what is obtained when running without any constraints, as we would expect. This data set exhibits a fairly high amount of spatial contiguity even when clustering without constraints (0.744, with $k = 4$). We observe what seems to be an inflection point in the graph around $s_{contig} = 0.0$, due to our non-linear constraint violation penalty. If we instead used a linear penalty, we would expect this graph to exhibit linear behavior. Instead, the non-linear penalty allows us to interpret a $|s_{contig}|$ of 1.0 as a completely hard constraint that cannot be violated.

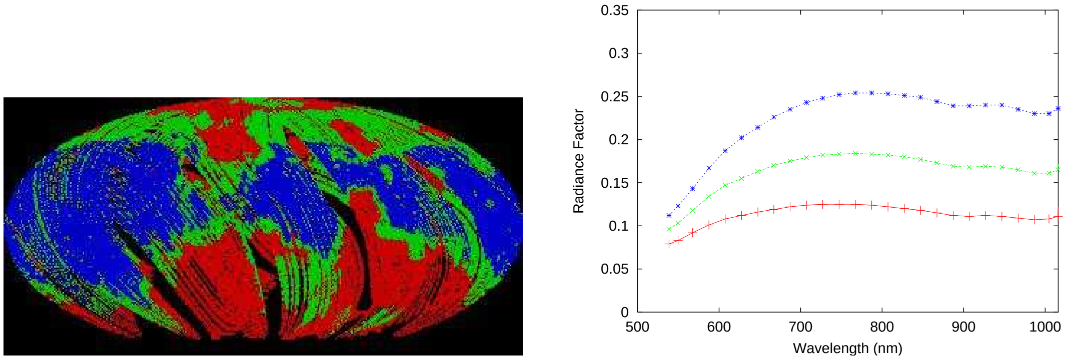
6.6.2 Comparison to Other Research on Mars

We have shown that SCOP-KMEANS can effectively make use of soft constraints by discussing our contiguity results. In addition, our analysis has also produced science results that largely agree with other investigations of Mars observations. Bell et al. (1997) collected HST observations of Mars in 1994 and 1995 at nine wavelengths (225–1042 nm) and analyzed the data using ratio images and scatterplots. They report three major findings:

1. **B1:** There are two distinct bright regions, largely distinguished by their $\frac{R_{673}}{R_{410}}$ ratios (one is 20–25% higher than the other).

2. **B2:** Dark regions have a 3–5% deeper absorption band at 860 nm than do the bright regions.

3. **B3:** The 953-nm absorption band has a depth of 0–5% for bright regions and a depth of 7–15% for dark regions.



(a) Classification (b) Average cluster spectra

Figure 6.5: K-means output for $k = 3$

Comparison to B1. Our observations do not include wavelengths as low as 410 nm, but we can examine the average values for FL for each cluster (this is the $\frac{R_{528}}{R_{762}}$ ratio). Running with $k = 3$ (and no constraints) produces clusters that correspond to one cluster covering the "dark" areas and two clusters collectively covering the "bright" areas (see Figure 6.5a). In the absence of constraints, we find that the two bright clusters have FL values of 0.523 and 0.446, which indicates that one has a ratio that is 17% higher than the other. The average cluster spectra are shown in Figure 6.5b. This agrees well with **B1**. It is important to note that these three clusters were identified based on their homogeneity; we did not identify "bright" and "dark" regions explicitly for the algorithm to analyze.

When performing the same analysis but incorporating a spatial contiguity constraint with strength 0.5, this result is unchanged; we still find two bright clusters with the same relationship. However, the clusters are slightly more contiguous (per-pixel contiguity rises from 0.802 to 0.814).

Comparison to B2. We observe less of a distinction between the bright and dark regions in terms of BD_{860} than was reported by Bell et al. Rather than the dark areas having a 3–5% deeper absorption band, we instead see a difference of only about 2%. However, the trend is the same.

Comparison to B3. Interestingly, our results directly contradict **B3**. Recall that the deepest band depth at 950 nm in our data set is only 6%. We observe a very minimal average 950-nm absorption band for bright regions and an absorption of 1–2% for dark regions.

There are two possible reasons for the discrepancy between our results and **B3**. First, we are using a very coarse division of the planet into two or three regions and labeling those as "bright" and "dark." Although the averages show very small absorption features, the constituent pixels in each cluster possess a variety of band depths. Pixels in the bright cluster have a BD_{950} that ranges from 4% to -7% (a negative band depth indicates a local reflectivity *maximum* rather than an absorption feature). Pixels in the dark cluster range from having a BD_{950} of 6% to -6%.

Second, we are probably not calculating the continuum ("expected") radiance value at 950 nm in the same way that Bell et al. did. This is important, because it directly affects the band depth calculation. This may explain why the maximum band depth we observed differs so greatly from their observations.

Regardless, we are encouraged by the degree of agreement we find between our independent analyses of the same planet. We find agreement even though the observations are separated by five years and some of the derived features have not been calculated in exactly the same way. In terms of observing conditions, Bell et al. noted that at the time they were collecting their data, Mars was experiencing very cloudy weather. In contrast, our data set was created during extraordinarily clear Martian conditions. This allows us to see more of the planetary mineralogy, due to the reduction in obscuring ice and clouds. As we will see in the next section, our automated analysis was also able to detect and isolate areas where ice and clouds are present.

6.6.3 Novel Scientific Findings

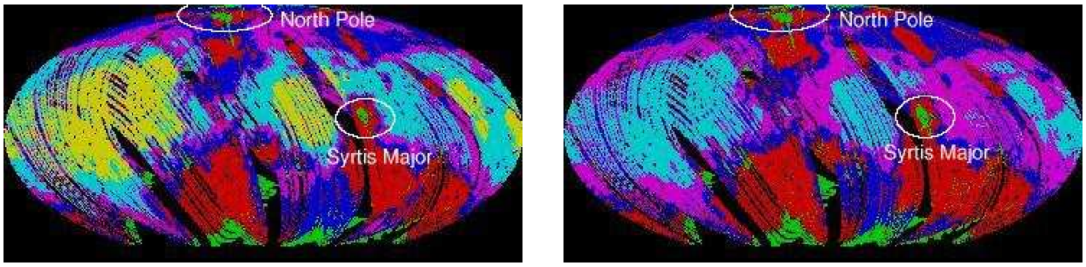
In addition to results that confirm or suggest alternatives to existing knowledge about Mars, our analysis also suggest several other interesting features of our particular data set. These observations add to the sum of our knowledge about the planet and are therefore valuable from a scientific viewpoint. In this section, we detail five interesting observations that came out of our automated analysis of the data set.

950-nm band depth. We found that this absorption band was a very distinct feature for regions on Mars. When clustering with $k = 2$, we find that the bright cluster has an average band depth of 0.3%. This is minimal, but it clearly contrasts with the average result for the dark clusters. Rather than finding an absorption feature at 950 nm, we find that the average value for BD_{950} is -2% (a local reflectivity maximum). This result suggests that the bright regions may be more likely to have a Fe^{3+} absorption band at this location.

800 to 1000 nm slope. This feature also turns out to be important for distinguishing regions on Mars, as we hypothesized. The overall average slope for this data set is negative from 800 to 1000 nm. Our results indicate that bright regions, on average, have a slope that is twice as strongly negative as the dark regions.

Unusually dark low-red regions. There are two regions on Mars that show up as having abnormally *dark and flat* spectra, in comparison with the rest of the planet. Most of Mars is covered with spectra that have low FL values; the average for our data set is 0.530. However, these two regions have an unusually high average FL value of 0.820 (at $k = 6$). In fact, every pixel in the cluster has an above-average FL value (from 0.642 to 1.018). These spectra, because they are so relatively flat, can also be considered "low-red" regions; they will appear much less red than most of the planet does.

Besides being flat, these spectra are also very dark, with very low overall radiance values. The spatial location of these pixels is relevant; they occur in two places, Syrtis Major and



(a) With contiguity constraints, $k = 6$, $s_{contig} = 0.5$

(b) Without constraints, $k = 5$

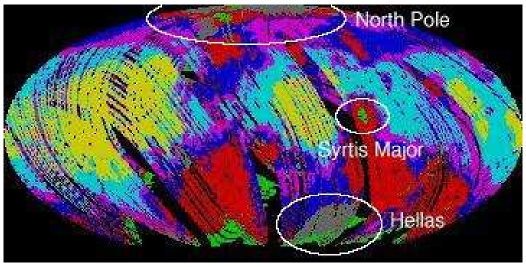
Figure 6.6: Dark low-red regions on Mars

two patches just southwest of the north polar cap. These are two places previously known to be very dark (McCord et al., 1971). Figure 6.6a shows the output of SCOP-KMEANS when using contiguity constraints with $k = 6$. The two regions of interest are labeled. (There are other pixels also assigned to this cluster, near the south edge of the planet. However, they are mostly artifacts of the data gathering process; they were observed under very poor conditions due to the viewing geometry between the Earth and Mars.)

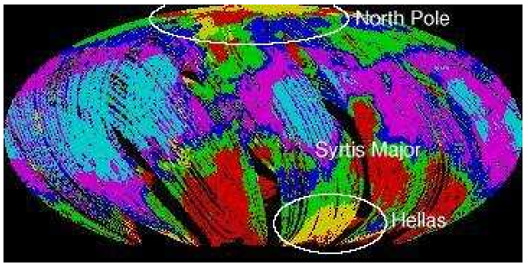
These results were obtained when clustering with contiguity constraints. If we do not use it, a similar cluster appears at $k = 5$, but it is much less distinguished; its average FL value is only 0.799. This, too, is above average, but it is less spatially (and spectrally) coherent. The variance for the cluster obtained when using the regular k-means algorithm is 0.068, while the variance of the corresponding cluster when using COP-KMEANSis only 0.057. Lower variance means that the average cluster spectrum is a better fit to the items in the cluster, spectrally. The average spectrum for the unconstrained cluster is shown in Figure 6.3b. It is the flattest cluster in the graph, clearly distinguished from the other clusters.

Ice/cloud regions. There are also two regions on Mars that are distinguished by *bright and flat* spectra. They appear in the north polar region and over the Hellas Basin. If we increase k to 7, this cluster appears with a FL of 0.745, which is higher than average yet distinguished from the dark, flat regions above. In fact, the higher value of k also permits a tighter fit to the dark flat cluster; it now has an average FL of 0.825. Because these regions are significantly brighter than dark flat cluster, and because of their spatial locations, we hypothesize that they represent ice (water ice or CO_2 ice) present at the pole and thin clouds present over the Hellas Basin. Both of these hypotheses are physically plausible and in accord with what we believe to be possible on Mars.

Figure 6.7a shows the output of SCOP-KMEANS when using contiguity constraints with $k = 7$. The north polar region is almost entirely covered by the ice/cloud cluster, as is the Hellas basin. The dark flat cluster is still visible (cf. Figure 6.6a). Once again, these are the results obtained when clustering with contiguity constraints. Without that information, the results are again less satisfactory. At $k = 6$, a cluster similar to the ice/cloud cluster appears (Figure 6.7b), but its FL value is at 0.751. We also observe that the ice/cloud



(a) With contiguity constraints, $k = 7$, $s_{contig} = 0.5$



(b) Without constraints, $k = 6$

Figure 6.7: Ice/cloud regions on Mars

cluster obtained when using contiguity constraints is a better fit to the data; that cluster's average variance is only 0.149, compared to an average variance of 0.178 obtained when clustering without constraints. In addition, without contiguity constraints the dark flat cluster disappears; some of its pixels appear to have been subsumed by the ice/cloud cluster, which explains why that cluster's FL value is higher.

Slope constraint has little effect. We experimented with different values for s_{slope} and different ways of calculating these constraints, but found that the constraints have little effect on the final results. This appears to be because the clusters found in the absence of constraints are already fairly well separated with respect to this feature; darker areas tend to have lower slope values than brighter regions, as mentioned above. It is possible that calculating this feature as a relative value rather than an absolute one might yield values that correlate less with albedo. We have not tested this hypothesis yet. Although the slope constraint did not significantly alter the results in this case, it is still a useful piece of knowledge to include. For exploratory analyses in particular, blind analysis of the data can produce misleading results. By including additional information such as the slope constraint, we can *guarantee* that the output will be reasonable in terms of this particular domain knowledge. Without such constraints, we cannot be certain that the results will be physically meaningful.

6.7 Summary

In this chapter, we have achieved two major goals. First, we have developed a formulation for soft constraints and a clustering algorithm that can make effective use of those constraints. Second, we applied this algorithm to a very large database of Mars observations and found that the results are of value scientifically both for confirming the results of independent analyses of the planet and for highlighting points of disagreement. These results indicate that constrained clustering can be a useful tool for scientific discovery. We believe that scientists from a variety of disciplines can benefit from adding constrained clustering to their set of frequently used analysis tools.