15-781: ML Project Final Report:  
Feature Selection for  
Rock Classification from Spectra

Jonathan Moody  Ricardo Silva  Joseph Vanderwaart
jwmoody@cs  rbas@cs  joev@cs

November 29, 2000

1 Introduction

Reflectance spectrometers have been used for identification of mineral composition of rocks and samples with varying degrees of success. This kind of spectrometer measures the amount of sunlight reflected by a rock or soil sample over a range of wavelengths. The reflectance obtained under different wavelengths can then be used to predict which minerals are present in that sample.

For instance, NASA intends to design robots for planetary exploration that would be sufficiently autonomous to interpret spectrometer data and report only the results back to Earth. Robots equipped with automatic classifiers of rock and soil samples would also be useful for automatically planning which different regions of a geological site would be more promising for prospecting certain classes of minerals.

The data sets collected by spectrometers consist of levels of reflectance intensity of a given rock at different wavelengths. The intensity data are typically measured relative to a reference surface in order to be invariant with respect to the total amount of sunlight in the environment.

The usual approach taken by someone interested in building a predictive model out of this data is running a regression model for each rock or soil sample, where the dependent variable is the reflectance intensity of the unknown rock and the independent variables are the reflectance intensities of a
variety of different pure minerals that are possible components of the rock, measured over the same wavelengths. Libraries of such pure mineral spectra exist; in particular, the Jet Propulsion Laboratory has produced a library of spectra for 135 different pure minerals, each containing reflectance intensities for 820 different wavelengths between 0.4 and 2.5 μm.

Assuming that the intensity of the rock is a linear combination of the intensity of its components, a regression model is built using each reflectance value at a wavelength as a data point. Then, only those minerals whose coefficients on the regression model pass a a given test of statistical significance are considered components of the rock. A successful learning algorithm should commit as few errors as possible, where a error is accepting a given mineral as part of a rock when this is not true, and rejecting a given mineral as part of rock when in fact it is.

Ramsey, et al, present evidence in [13] that a specialized Bayes net construction algorithm (the PC algorithm) performs better than simple linear regression for classifying carbonates. The PC algorithm tests partial correlation between an input spectrum and different subsets of the library spectra, eliminating library spectra (hypothesized components) which are not correlated with the input. The remaining library spectra are assumed to be components of the input, and a a classification can be inferred. All of our work described below is based on this classification algorithm.

2 Description of the problem

Experiments with the specific class of carbonates have shown that restricting the input of the PC algorithm to a region of the spectrum can improve accuracy. In particular, a region suggested by prior expert knowledge (a region used by experts to identify carbonates) produces much better results than allowing the algorithm to consider entire spectrum. This is a promising result that arguably can be extended to other classes.

However, coming up with a interesting range of intervals is not a easy task, and little is known for other minerals. No automated method has been applied by the authors of [13] to find subintervals that would be more appropriate for identifying given classes and subclasses of minerals.

Our goal was to find intervals of the spectrum, specific to the class of minerals, for which the PC algorithm performs better than the same algorithm using the entire spectrum. This is an instance of the feature selection
(alternatively, data cleaning) problem in machine learning. We tried several methods, a collection representative of both the “wrapper” and “filter” approaches as described in [8].

3 Data selection techniques

Finding an appropriate subset of the spectrum range can be cast as a problem of search among the space of possible subsets. Since we have over 800 available channels, an exhaustive search is infeasible. Also, a larger number of evaluated candidates increases the chance of overfitting [2]. One must decide how to trade-off the complexity of the search space depending on the chosen search algorithm, on the available computational resources and amount of data. Within the wrapper framework, three algorithms were tried: a computationally demanding genetic algorithm and two greedy hill-climbing algorithms.

A more straightforward approach would be to construct a “relevance” heuristic, rank the channels accordingly, and select those with relevance above a threshold. Intuitively, we wish to discover those channels that carry a large amount of information relevant to the question of whether a certain class of minerals is present. Therefore, we used information gain, a quantity based on entropy, for our relevance heuristic. Information gain is commonly used to construct decision trees [10].

3.1 Genetic algorithm

A genetic algorithm (GA) is an algorithm for combinatorial optimization [5], which is directly related to the task of finding useful subsets of the spectra. The most straightforward representation of a candidate is through a string of 826 bits, where a positive bit represents that the respective channel will be used. However, due to the reasons explained in the beginning of this section, we divided the spectrum into a fixed number of blocks, each represented by a bit. Thus, all channels in the same block are selected or not selected at the same time.

The evaluation function is very time-consuming, consisting of running the modified PC algorithm over a whole set of rock samples. The fitness of a candidate is the proportion of rocks that are correctly classified as containing or not containing the respective mineral. On our available implementation,
it takes about 30 seconds to evaluate a single candidate on a Pentium III 733MHz processor.

3.2 Bitwise hill-climbing

We also used a greedy, hill-climbing algorithm that uses the same representation for search states and the same evaluation function. On the initial state, all bits are activated. The next states are the generated from the current state by setting to zero one of the currently activated bits. If the current candidate has $n$ activated bits, it will generate $n$ new candidates. The candidate with the highest evaluation value is chosen to be the next state.

3.3 “Peeling” algorithm

This is another greedy algorithm that is also used for rule induction over continuous/ordered attributes [4]. It consists of trimming the extremes of an interval by some percentage of the data and evaluating the new interval obtained. A typical strategy starts with the complete interval and, at each subsequent step, generates three new candidates: the current interval with the bottom $\alpha\%$ of the ordered data discarded, the current interval with the upper $\alpha\%$ of the ordered data discarded, and an interval constructed by dropping the bottom and upper $\frac{\alpha}{2}\%$ from the current interval.

The underlying assumption of this algorithm is that interesting intervals are continuous. Unlike the previous algorithms, all selected subintervals are of the form $[a, b]$, where $a$ and $b$ are points of the original interval. It may clearly result in suboptimal selections, at the advantage of being much less time demanding.

3.4 Information gain heuristic

The information gain algorithm for selecting a channel mask is as follows. For each channel, we divide the intensity range into some number of bins (for our experiments, we used four, but this number could be varied). Then for every spectrum in the reference library we look at the intensity at the current channel and take note of which bin it occupies and whether or not it is a member of the target class. When we have finished doing this for a given channel, we calculate the fraction of samples in each bin that are in the desired class; this number is used to calculate an entropy value for that
bin. A weighted sum of the entropies of the bins gives the expected entropy
given the intensity of a particular channel; subtracting this from a constant
gives the expected information gain associated with that channel.

When we have calculated the expected gain for each channel, we create
a channel mask by looking for intervals where the expected gain is higher
than average. Specifically, we divide the spectrum into blocks and calculate
the average expected gain in each block. Then blocks whose average ex-
pected gain exceeds the global average by some margin are selected for use
in classification.

An important difference between this approach and the others we at-
tempted is that it is not strictly speaking a search algorithm and therefore
does not really need separate training and test data. We did, however, need
to be able to experiment with different values for certain parameters of the
algorithm (in particular the number of bins, the size of the blocks of channels
and the threshold value) without corrupting our results due to overfitting.
We performed these experiments on one half of the dataset, and then tested
the best parameter settings on the other half.

4 Experiments

For our experiments we used the NASA Jet Propulsion Laboratory (JPL)
data set as a reference library, attempting to classify the rocks in the Johns
Hopkins Univ. (JHU) data set. For further information on these data sets,
see Ramsey, et al. [13].

We performed experiments using four of the mineral classes available in
the JPL library. These minerals were chosen according to the number of
rocks present in the JHU data set that were reported to have these minerals:
it would be unreliable to try to find intervals for a class underrepresented on
the available data. Among all 192 JHU rocks, 92 have carbonates, 121 have
phyllosilicates, 100 have oxides and 84 have inosilicates.

Tables 1 and 2 show the results for running the modified PC algorithm
using the intervals selected by variations of each algorithm described on the
previous sections. For each mineral class, the JHU data set was split into
two data sets, A and B, each with half of the complete data. In splitting the
JHU data, we were careful to balance the number of positive and negative
examples in the training and testing sets. We also selected splits which gave
similar classification accuracy over the entire spectrum, in an effort to assure
that one of the data sets was not vastly more difficult to classify than the other. A 2-fold cross-validation was performed with these sets. We opted for only 2 folders because:

- the genetic algorithm is computationally intensive;
- we wanted a reasonable amount of data on both training and test sets. Using a high number of folders can in fact lead to worse generalization estimates when we have few data points and the accuracy error is high, as it is typical of this domain [14].

Two variations of the genetic algorithm were used, one with 15 genes (GA-15) and other with 10 (GA-10). A higher number of blocks enables the selection of more complex subintervals, but also introduces more free parameters. The choice of 15 genes and 10 genes was also influenced by the amount of time consumed on the computation, since the higher the number of parameters, the higher the number of individuals one would like to use to reduce the chance of being stuck on a local minima too early. For GA-10, we used 25 individuals and for GA-15, 35 individuals. The training proceeded for 25 and 40 generations respectively. In all cases, by the last generation the pool of individuals was almost completely dominated by copies of a single individual (and in many cases, all individuals were identical), suggesting that further optimization would not improve the result obtained significantly. The code of the genetic algorithm was adapted from [9], with its default parameters.

We also used cached statistics to scale up the algorithm: instead of passing through all the data points when computing an element of the correlation matrix (as required by the PC algorithm), we precomputed the summations and inner products of variables for the data falling under each block. Getting a new element of the correlation matrix required only a pass over these cached statistics. This procedure reduced the computational time by over 30%.

Two variations of the block-based hill-climbing search were also used: H-10 and H-20, with 10 and 20 blocks respectively. As a trade-off to avoid bad local maxima without searching till the last state, the search stopped when we did not get improved results by five consecutive states. The best selection on this search path was the output.

With the peeling algorithm, we used a 5% $\alpha$. We used the same stop criterion applied on the previously described hill-climbing technique.
For our experiments with the information gain heuristic, we chose to
divide the intensity range into 4 bins for the purpose of calculating the en-
tropy. To make it easy to qualitatively compare the masks generated by
this heuristic with the ones found by the searching algorithms, we performed
experiments selecting the channels in blocks of 54 and in blocks of 80 (this
produced 16-bit and 11-bit masks, respectively).

The most difficult decision in calibrating the gain heuristic turned out
to be the selection of a threshold gain value to use when selecting blocks of
channels. After some preliminary experimentation, we decided on a threshold
of 0.4 standard deviations above the global mean gain. That is, we calculated
the mean μ and standard deviation σ of the gains of all the channels and
selected blocks whose average gain was at least (μ + 0.4σ).

The results of our information gain experiments are displayed in Tables
1 and 2. As that table shows, the results are mixed. While our performance
on carbonates and inosilicates improved relative to the baseline of enabling
all channels, it was worse on phyllosilicates and oxides. In fact, only one of
the experiments shown in the table was able to produce a mask that allowed
oxides to be correctly classified more than half the time, and two of the others
were worse than the PC algorithm with all channels enabled.

Interestingly, the accuracy attained with the information gain heuristic
was extremely sensitive to the parameters of the algorithm, namely number
of bins, block size and threshold gain value. By trying different settings for
these parameters, we were able to produce some masks that outperformed
the ones shown on both halves of the data set; however, because the data
set is so small we suspect that these “better” results may have been due to
overfitting and thus might not work well when applied to new data.

None of the algorithms performed significantly better on average than
the expert choices for carbonates, which results in 66.7% accuracy for data
set A and 68.8% for data set B, but most of them were able to improve
over the complete spectrum. Results for oxides and phyllosilicates reflected
the difficulty of this domain. This is a ill-defined problem, and it is expected
that for some classes the reflectance data is not enough for good classification
of rock samples. Among the algorithms that we applied, H-10 got the best
result for phyllosilicates.

For inosilicates, the genetic algorithms and information gain metrics got
the best average. These two variations also performed well for carbonates,
with H-10 being not very far.

The most visible problem we faced was sampling variation. Results using
Table 1: Accuracy on set (B) (using set (A) for feature search)

<table>
<thead>
<tr>
<th></th>
<th>None</th>
<th>GA-10</th>
<th>GA-15</th>
<th>H-10</th>
<th>H-20</th>
<th>PEEL</th>
<th>IG-54</th>
<th>IG-80</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carbonates</td>
<td>56.3</td>
<td>65.6</td>
<td>70.8</td>
<td>60.4</td>
<td>60.4</td>
<td>52.1</td>
<td>64.6</td>
<td>64.6</td>
</tr>
<tr>
<td>Inosilicates</td>
<td>61.5</td>
<td>76.0</td>
<td>80.2</td>
<td>58.3</td>
<td>59.4</td>
<td>64.6</td>
<td>71.9</td>
<td>70.8</td>
</tr>
<tr>
<td>Oxides</td>
<td>47.9</td>
<td>50.0</td>
<td>50.0</td>
<td>53.2</td>
<td>53.6</td>
<td>43.8</td>
<td>50.0</td>
<td>44.8</td>
</tr>
<tr>
<td>Phyllosilicates</td>
<td>55.8</td>
<td>54.7</td>
<td>51.6</td>
<td>60.0</td>
<td>52.1</td>
<td>55.8</td>
<td>57.9</td>
<td>50.5</td>
</tr>
</tbody>
</table>

Table 2: Accuracy on set (A) (using set (B) for feature search)

<table>
<thead>
<tr>
<th></th>
<th>None</th>
<th>GA-10</th>
<th>GA-15</th>
<th>H-10</th>
<th>H-20</th>
<th>PEEL</th>
<th>IG-54</th>
<th>IG-80</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carbonates</td>
<td>56.3</td>
<td>62.5</td>
<td>64.6</td>
<td>69.8</td>
<td>58.3</td>
<td>40.1</td>
<td>62.5</td>
<td>61.5</td>
</tr>
<tr>
<td>Inosilicates</td>
<td>60.4</td>
<td>65.6</td>
<td>66.7</td>
<td>65.6</td>
<td>67.7</td>
<td>59.3</td>
<td>65.6</td>
<td>63.5</td>
</tr>
<tr>
<td>Oxides</td>
<td>47.9</td>
<td>56.2</td>
<td>46.9</td>
<td>51.0</td>
<td>53.1</td>
<td>53.1</td>
<td>45.8</td>
<td>54.2</td>
</tr>
<tr>
<td>Phyllosilicates</td>
<td>56.7</td>
<td>55.6</td>
<td>52.6</td>
<td>58.8</td>
<td>49.5</td>
<td>54.6</td>
<td>54.6</td>
<td>48.5</td>
</tr>
</tbody>
</table>

data set A for training, instead of B, were sometimes significantly different. However, it is interesting to note that for carbonates and inosilicates, the rank from the best to worst performing algorithm was largely the same, excepted for the H-10 and H-20 algorithms, which showed the largest instability. Overall, the genetic algorithm with 15 genes performed better for these classes.

The selected intervals for GA, H and IG are shown on Tables (3,4) of the appendix. The intervals obtained by the two GA experiments are more consistent than the respective H results when comparing the output for data sets A and B, but they still presented considerable variation.

5 Related work

The techniques applied in this work are related to the areas of feature selection and data cleaning. Wetschereck, Aha and Mohri [16] formulate a framework for feature weighting methods under the context of lazy learning.\(^1\) Even though in a strict sense the wavelengths channels are in fact rows of our data set, not attributes, in principle one can use these techniques to

\(^1\)On this survey, the authors do not compare different batch optimization techniques: among this class of learning algorithms, only a gradient-based one is used.
weight the relevance of each data point (or intervals for practical purposes). According to the categories of Wettschereck et al's framework, the genetic algorithm and hill-climbing approaches would be classified as having:

- a performance bias, since we use the actual results of classification for deciding the selection;
- a binary weight space (i.e., 0/1 weights);
- a transformed representations, since we divide the data into blocks;
- a global weighting, because the same intervals are selected for all minerals;
- knowledge-poor, since we did not use prior knowledge in our experiments;

The performance bias is also commonly described as a wrapper approach [8]: our selection policies use the modified PC algorithm as a black box that outputs a measure of performance (figure 1).

Unlike general feature selection problems, we do not have the concern of selecting features that present less missing values on the available data bases, nor do we have to consider which are more expensive to measure (e.g., some medical exams for diagnosis problems). That makes our fitness function even simpler than most ones used on feature selection literature [11, 15, 17]. These approaches are virtually identical to the genetic algorithm for data selection described in this work, where the difference is mainly a more complicated evaluation function. Demiroz and Guvenir [3] also describe mechanisms for learning continuous weights between 0 and 1, which arguably are not much useful to our problem, where we have little data to generate such a solution that requires a more precise tuning of parameters.

In contrast, the information gain heuristic operates as a filter approach (figure 2). The filter approach applies for each feature a measure of importance that is independent of the learning algorithm that will be used. Hall [7] provides a comparison of filters and wrappers, as well as an overview of feature selection. He favors the filter approach due to its much higher scalability, but in his discussion it is mentioned that ideally the features themselves should be a function of the bias of the learning algorithm that will be used. An intermediate approach such as using the entropy measurements to search
Figure 1: the “wrapper” approach

Figure 2: the “filter” approach
for a combination of prominent intervals, which can then be successfully used by the modified PC algorithm, is a way to trade-off these issues.

Entropy measures are commonly related to the degree of unexpectedness of a pattern, and such a characteristic has been explored for data set cleaning. Guyon, Matic and Vapnik [6] describe different ways of using information theoretical measures to identify outliers or highly informative examples. Data points are ranked according to information gain and then submitted to an expert that will classify them as outliers or representative examples. Guyon et al. warn against the risk of getting improved results during training by dropping the most difficult examples and then achieving bad generalization accuracy.

Another application of information theoretical measures for data cleaning is discussed by Pyle [12], where it is also described how to find ill-defined regions of a function by checking symmetries between the input and output variables. This specially affects inverse function estimators. Pyle also describes what he calls “attention processing” of data: how to perform data surveying and avoiding on the combinatorial explosion of the search space of potentially problematic regions of the data.

6 Conclusions and future work

The experiments provided some evidence that approaches such as genetic algorithms can improve the classification performance for this domain. However, sampling variability is a concern. We encountered large (up to 13%) variations between performance on set A and set B. [8] reports that feature selection algorithms may overfit easily. Approaches to minimize this problem and perform more reliable perform assessment include resampling techniques such as bootstrapping [1].

This improved reliability does not come for free, and more computational time is required. For instance, Punch et al. [11] reported experiments with genetic algorithms for feature selection that took 14 days. In this case, one might not want genetic algorithms, since the difference in accuracy when compared with other approaches may not be great enough to justify the extra effort. Alternatively, one could just gather more labelled data. For example, the U.S. Geological Survey produced a data set of about 400 labelled rocks. However, some of these labels are wrong, or inconsistent with the class names of the JPL data set.
There are two major directions for future work on the information gain algorithms. First, it would be interesting to use rock spectra (as opposed to pure mineral spectra from the reference library) when computing the per-channel gain. This would correspond more directly to the intuition that we should select channels that are informative with respect to identifying rocks that contain a certain class of minerals, rather than selecting channels that are informative with respect to discriminating between different types of pure minerals as we have done in this study.

Second, more sophisticated ways of extracting channel selection masks from the entropy data should be investigated. Selecting a subset of a fixed partition of the channel set may limit the flexibility of the algorithm to include all the channels that are helpful while discarding all the of the ones that are confusing; other ways of deciding on high-gain subintervals might allow better masks to be found.

References


7 Appendix

<table>
<thead>
<tr>
<th>Feature Mask</th>
<th>Carboate</th>
<th>Iodinate</th>
<th>Oxide</th>
<th>Phosinate</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA-10</td>
<td>0001001111</td>
<td>1101011000</td>
<td>0010111111</td>
<td>1111100110</td>
</tr>
<tr>
<td>CA-15</td>
<td>000000000000001</td>
<td>10000000111110000</td>
<td>00000001111110010</td>
<td>101011111110000</td>
</tr>
<tr>
<td>B-10</td>
<td>000000000000001</td>
<td>00010110111</td>
<td>00010110001</td>
<td>11001110110</td>
</tr>
<tr>
<td>B-20</td>
<td>000000000000001</td>
<td>11111111111110111</td>
<td>11101000000101111</td>
<td>11101000111110000</td>
</tr>
<tr>
<td>B-35</td>
<td>000000000000001</td>
<td>00100001110000000</td>
<td>000000000001111000</td>
<td>00000000000111100</td>
</tr>
<tr>
<td>B-30</td>
<td>000000000000001</td>
<td>00100010000000</td>
<td>000000011110000</td>
<td>000000000001111</td>
</tr>
</tbody>
</table>

Table 3: Feature masks obtained by training on data set (A) (except for IG, which does not require training)

<table>
<thead>
<tr>
<th>Feature Mask</th>
<th>Carboate</th>
<th>Iodinate</th>
<th>Oxide</th>
<th>Phosinate</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA-10</td>
<td>0001001111</td>
<td>1101011000</td>
<td>0010111111</td>
<td>1111100110</td>
</tr>
<tr>
<td>CA-15</td>
<td>0111111100000111</td>
<td>110101101101100</td>
<td>000101111101111</td>
<td>101111111101100</td>
</tr>
<tr>
<td>B-10</td>
<td>01100000001</td>
<td>01101101101</td>
<td>01101000111</td>
<td>11010111130</td>
</tr>
<tr>
<td>B-20</td>
<td>11111101101100111</td>
<td>10100000011110000</td>
<td>01100110101111111</td>
<td>00010001111110111</td>
</tr>
<tr>
<td>B-35</td>
<td>000000000000001</td>
<td>001000011110000</td>
<td>000000000001111000</td>
<td>00000000000111100</td>
</tr>
<tr>
<td>B-30</td>
<td>000000000000001</td>
<td>00100010000000</td>
<td>000000011110000</td>
<td>0000000000011110</td>
</tr>
</tbody>
</table>

Table 4: Feature masks obtained by training on data set (B) (except for IG, which does not require training)