DESIGN TRADEOFFS IN A HARDWARE IMPLEMENTATION OF THE K-MEANS CLUSTERING ALGORITHM

Miriam Leeser[1], *James Theiler*[2], *Michael Estlick*[1], *and John J. Szymanski*[2]

[1]Department of Electrical and Computer Engineering
Northeastern University, Boston, MA 02115
mel@ece.neu.edu, mestlick@ece.neu.edu

[2]Space and Remote Sensing Sciences Group,
Los Alamos National Laboratory, Los Alamos, NM 87545
jt@lanl.gov, szymanski@lanl.gov

ABSTRACT

Hyperspectral imagery provides exquisitely detailed information, but poses a serious challenge to the image analyst. Massive quantities of data must be reduced in a way that identifies useful features. Pixel clustering algorithms provide one approach: each pixel is assigned to a class based on the spectral similarity of that pixel to other members of the class. An image where each pixel is represented by its class provides a single picture (not hundreds of channels) for the analyst to interpret.

Opportunities for fine-grain parallelism in this computationally intensive calculation motivate the consideration of specialized hardware. We are developing an FPGA-based implementation of the k-means clustering algorithm, because FPGAs can provide considerable speedup yet are sufficiently flexible to permit the testing of variants and tradeoffs.

Assigning pixels to classes requires a definition of spectral similarity. The standard choice is the Euclidean distance (sum of the squared differences between the pixel and the cluster center taken on a coordinate by coordinate basis). We consider two alternative distances: the Manhattan (sum of the absolute coordinate differences) and the Max (maximum of the absolute differences). The Euclidean distance is optimal in the sense of minimizing within-class variance, but requires computing many squares. The Manhattan and Max distances eliminate this squaring, and furthermore reduce the number of bits needed to store intermediate calculations.

We conducted two sets of experiments to assess the tradeoff between quality of clustering and efficiency of hardware implementation. The first considered an idealized setup for assessing mis-classification rates and their effect on within-class variance. The second evaluated within-class variance for real 224-channel AVIRIS data sets. Both experiments confirmed that the highest quality clusters are achieved using the Euclidean metric, and quantified the difference. In comparing the faster, alternative distances, we found with the AVIRIS data that the results depended on which image was used; however, the idealized experiment showed a clear preference for the Manhattan metric.

1. INTRODUCTION

Hyperspectral images, with upwards of 200 spectral channels, are an increasingly common data product in remote sensing applications. These data provide important and useful information, but the challenge for the data analyst is to identify the important and useful features without being overwhelmed by the sheer volume of the data. One approach is provided by algorithms which segment the image by clustering pixels into classes, based on the spectral similarity of each pixel to other members of the class. This can represent a massive, albeit lossy, compression of data. For instance, we have been working with AVIRIS data [1], in which each 614x512 pixel image has 224 16-bit channels; about 140 MB of data. When clustered into 16 classes, each pixel is represented by a 4-bit pointer to its class. The resulting image is about 150KB – for a compression factor of three orders of magnitude.

As well as reducing the data for quicklook views, clustering also provides an organization of the data that can be useful for other processing downstream. Several authors have shown that clustering the data beforehand increases the performance of algorithms which attempt to "learn" features from a small number of examples [2, 3]. Schowengerdt [4] suggests the use of image segmentation for change detection: a change in the segmentation is more likely to indicate an actual change on the ground, since the segmentation is relatively robust to changes in sensor performance and atmospheric conditions. It has also been argued that signal-to-clutter ratios can be improved by treating individual clusters separately; the variance within a cluster is generally much smaller thant the variance over the whole image.

While there are clear benefits to clustering high-dimensional data sets, workstation implementations of clustering algorithms can be slow. Most algorithms, such as k-means, are iterative and require numerous passes through the data before convergence is achieved. Each iteration requires a computation of distance from every data point to every cluster center, and each distance requires a loop over all the spectral channels. To accelerate this computation, we are developing an FPGA (Field Programmable Gate Array) implementation of the k-means algorithm. Reconfigurable hardware provides the ability to exploit the fine-grained parallelism inherent in the computation, while maintaining the flexibility to consider variants on the design and on the algorithm. FPGAs are particularly well suited to this application because the amount of parallelism and processing element bit widths can adapt to the task, allowing the designer to take maximum advantage of the hardware at hand. In addition, clustering on an FPGA board frees the workstation for the analyst to apply other algorithms to the data.

Implementations in hardware require a different set of design tradeoffs than implementations in software. For example, the main goal of a software implementation may be to minimize the number of iterations. In hardware, our goals are to simplify the underlying operations as much as possible in order to speed up the calculations and to be able to provide more parallelism. A simpler operation translates to a smaller area datapath which in turn translates to more versions of the datapath replicated on the chip. To this end, we are investigating alternative distance measures. By using a Manhattan or a Max distance, instead of the more expensive Euclidean distance, we trade theoretical optimality for practical efficiency. The Manhattan and Max distance calculations both eliminate a multiplier, thus saving area. They also reduce the number of bits needed to store intermediate calculations, minimizing the area required for routing.

This paper represents a set of experiments to compare Euclidean and non-Euclidean metrics for assigning data to a cluster. We conduct two experiments. The first, a data-independent experiment, shows an idealized setup for assessing mis-classification rates and their effect on the total within-class variance. The second set of experiments will show how Euclidean vs. Manhattan distance effects total variance for some AVIRIS data sets.

2. K-MEANS CLUSTERING

Given a set of N pixels, each one representing D spectral channels, and represented as a point in D-dimensional Euclidean space (that is, $\mathbf{x}_n \in \mathcal{R}^D$, with $n = 1, \ldots, N$); we partition the pixels into K clusters with the property that pixels in the same cluster are spectrally similar. The k-means clustering algorithms (there are several variants) provide an iterative scheme that operates over a fixed number (K) of clusters, while attempting to satisfy the following properties:

1. Each class has a center which is the mean position of all the samples in that class.
2. Each sample is in the class whose center it is closest to.

From an initial sampling, the algorithm loops over all the data points, and reassigns each to the cluster whose center it is closest to. After the pass through the data, the cluster centers are recomputed. Note that other variants of k-means update the cluster centers each time a point is reassigned to a new cluster. This leads to faster convergence, but is more difficult to implement in hardware. Each iteration reduces the total within-class variance for the clustering, so it is guaranteed that after enough iterations, the algorithm will converge, and further passes will not reassign points. The converged clustering will satisfy properties 1 and 2 above, but it is not necessarily (or even usually) the only partition of the data that satisfies these properties.

The k-means algorithm was originally designed to minimize the average within-cluster variance of a clustering.

The results of the k means algorithm depend on the method used for initialization. A common method of initialization is to randomly assign pixels to classes, then calculate the means of the resulting classes. This technique generates an initial set of means which are in general very close to one another. For this set of experiments we use a hierarchical initialization method, where k-means is run on a small set of pixels picked at random from the data set. k-means is run on this smaller set of data, and the resulting cluster centers are used to initialize the k-means clustering on the full data set.

3. NON-EUCLIDEAN DISTANCE MEASURES

Points are assigned to the cluster centers to which they are closest. Consider a point $\mathbf{x}$ and cluster center $\mathbf{c}$ where i indexes the spectral components of each. The Euclidean distance is defined as:

$$\|\mathbf{x}-\mathbf{c}\|^2 = \sum_i |x_i - c_i|^2. \tag{1}$$

But other distance measures can also be used; for instance, the general family of p-metrics (for which the Euclidean distance is the special case $p = 2$):

$$\|\mathbf{x}-\mathbf{c}\|^p = \sum_i |x_i - c_i|^p. \tag{2}$$

To perform a k-means iteration, one must compute the distance from every point to every center. If there are N points, K centers, and D spectral channels, then there will be $O(NKD)$ operations. For the Euclidean distance, each operation requires computing the square of a number.

The Euclidean distance has several advantages. For one, the distance is rotationally invariant. Furthermore, minimizing the Euclidean distance minimizes the within-class variance. On the other hand, the Euclidean distance is more expensive than two alternatives that we considered as compromises. The Manhattan distance, corresponding to $p = 1$, is the sum of absolute values of the coordinate differences; the Max distance, corresponding to $p = \infty$ is the maximum of the absolute values of the coordinate differences. Neither distance requires any multiplication, and the Max distance has the slight advantage that the number of bits required to express the total distance does not increase with the dimension D (for the Manhattan distance, we require $\log_2 D$ extra bits for the total distance over and above the bits required for the coordinate differences).

4. DATA INDEPENDENT ASSESSMENT OF NON-EUCLIDEAN METRICS

In assigning a data point to a cluster center, we assign to the center which is closest according to one of the cheaper metrics. We expect that in general closeness in the Manhattan or Max metric implies closeness in the Euclidean metric, but we recognize that this is an approximation that may lead to non-optimal clusterings. See Figure 1.

In assessing this tradeoff, we performed an experiment to estimate how often points would be mis-assigned because a cheaper distance metric was used. The effect of metric choice on the quality of a clustering depends on many factors: the number of clusters, the dimension of the space, and the nature of the data in that space. To eliminate as many of these factors as possible, we did a Monte-Carlo experiment with 10^5 trials. Each trial involved placing three points at random on the surface of a D-dimensional sphere. Two of the points were taken to be cluster centers, and the third was a data point. We computed the distance from the data point to the two centers, using all three metrics (Euclidean, Manhattan, and Max), and noted whether the cheaper metrics assigned the point correctly; *i.e.,* to the center that was closest according to the Euclidean metric.

The results are shown in Figure 2 as a function of the only free parameter in the experiment, the number D of spectral channels. As the dimension D increases, the mis-classification rate also

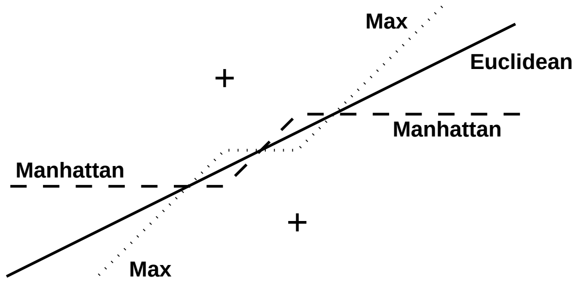

Figure 1: Two cluster centers are indicated by the two crosses in the figure above. The solid line partitions the area into points closer to the top cross and those closer to the bottom cross, where closeness is defined in terms of the Euclidean distance metric. If distance is defined in terms of a Manhattan metric, then the partition is given by the dashed line; if distance is given by the Max metric, then the partition is given by the dotted line. Points in the figure that are above the solid line, but below the dashed line will be misclassified by the Manhattan metric as belonging to the bottom class, when in fact they would be assigned by the Euclidean metric to the top class.

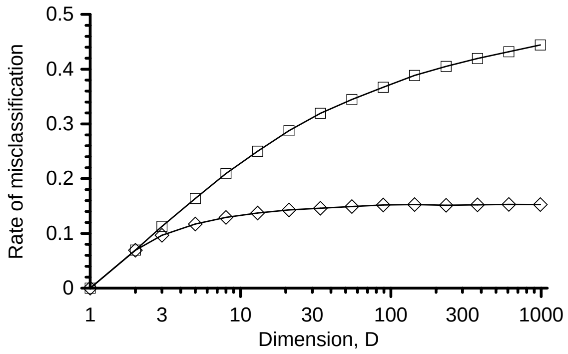

Figure 2: Probability of misclassification for $K = 2$ classes, plotted against the dimension D of the sphere onto which a triplet of points have been randomly placed. Two of the points in each triplet represent cluster centers, and the third a random pixel. A misclassification occurs when the center that is closest according to the Euclidean metric is not closest according to one of the other metrics. The diamonds represent the Manhattan metric, and the squares represent the Max metric. The misclassification rate increases with the dimension of the space; it saturates at about fifteen percent for the Manhattan metric, but is increasing toward one-half for the Max metric.

increases. For the Manhattan metric, this rate saturates at about fifteen percent, but for the Max metric, the error rate begins to approach fifty percent.

Another measure of error is the contribution to the total within-class variance. If a data point is nearly equidistant to two cluster centers, then a mis-classification does not incur a significant penalty. Figure 3 shows the relative effect on the within-class variance of using the different metrics. As the dimension becomes very large, the total variance is so large, that the relative effect decreases. For the Manhattan metric, the relative increase peaks at two percent; for the Max metric, it is always less than five percent.

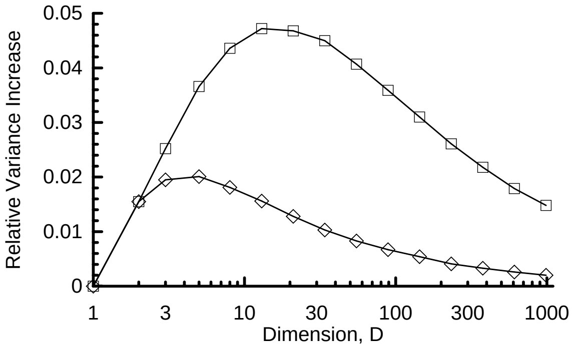

Figure 3: The effect of misclassification is to increase the within-class variance. This increased variance for the $K = 2$ class clustering problem is plotted against the number D of spectral channels. It is the relative variance increase (that is, increased variance divided by total variance) that is plotted, both for the Manhattan (diamonds) and for the Max (squares) metric. While the previous figure showed that the rate of misclassification increases monotonically with dimension, this figure shows that the effect of this misclassification begins to decrease for large dimension D.

We also performed this experiment for $K = 16$ classes. In this case, a trial consisted of 17 points placed at random on a D-dimensional sphere. Sixteen of the points were considered cluster centers, and the seventeenth point was the pixel whose class was assigned according to which of the 16 centers it was closest. The results of a Monte-Carlo experiment with 10^5 trials are shown in Figure 4.

While it is difficult to relate these idealized experiments to quantitative results on real data, they do provide indications about the relative performance of the two distance metrics. In particular, it is clear that the Manhattan distance results are closer to the cluster quality of the Euclidean metric than the Max metric is.

5. EXPERIMENT ON AVIRIS DATA

The primary goal of this experiment was to determine if the Manhattan and/or Max distance measures can be used in the k-means algorithm to acceptably cluster AVIRIS images, compared to using the Euclidean distance measure. We also considered the effects of using fractional bits in the cluster center data for each of the three distance measures: Manhattan, Euclidean, and Max.

The experiment was performed with five AVIRIS images and a C implementation of the k-means algorithm. The C implementation allowed us to set the following parameters: number of clusters, number of iterations, distance measure. The C implementation used simple floating point arithmetic. A second implemen-

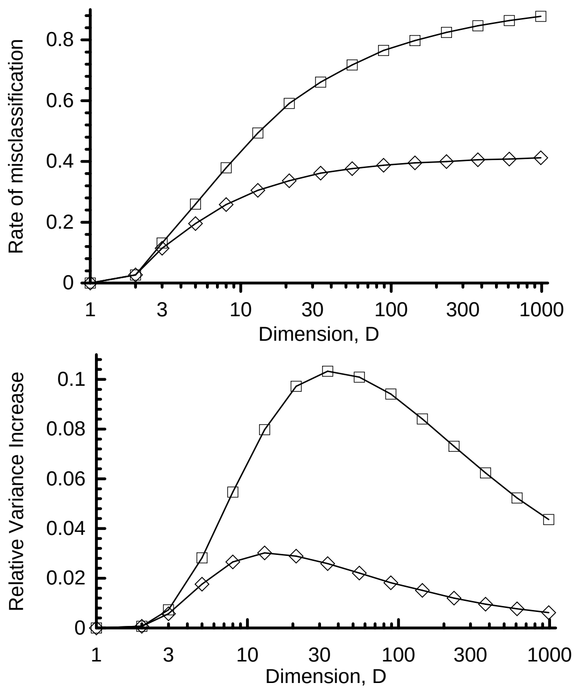

Figure 4: Misclassification and Relative Variance Increase for clustering into $K = 16$ classes. Both misclassification rate and relative variance increase are larger for this case than for the $K = 2$ case, but even with the larger number of classes, the effect of the alternative distance metrics is bounded.

tation in C++ also allowed us to set the number of fractional bits used in cluster center data. The C++ implementation used 64 bit integers for accumulating values, because the amount of data in a AVIRIS image would overflow a 32 bit integer. Integers were used in this implementation, because the hardware implementation will use integers. The fractional bits in the cluster means were obtained by shifting the accumulated pixel values before dividing by the number of pixels in the cluster. For both implementations, the output of a classification is a text file and an image file. The text file contains the number of pixels moved and the total variance for each iteration of the classification. The image file shows which clusters pixels were assigned to.

Two approaches were considered for initializing the k-means classifications. The simplest divides the image into K equal width "stripes" so that the first N/K points comprise the first cluster, the next N/K points comprise the second cluster, *etc.* We also investigated a potentially more efficient hierarchical initialization. For that initialization, we choose 256 pixels at random from the AVIRIS image, and randomly assigned each of the 256 pixels to a cluster. The k-means algorithm is then run until convergence. The resulting cluster centers are then used as the initial cluster centers for the whole image. The initial hierarchical classification is always done using the Euclidean distance measure, because it will always be done on a general purpose computer to initialize the hardware. If a cluster ends up with no pixels in the initialization, the cluster means are set equal to the means of the cluster with the largest variance, and then perturbed by ± 1 randomly. If a cluster ends up with no pixels in the classification of the whole image, the cluster is marked dead and not used any more. In practice, using these initialization schemes, we've not seen dead clusters in any of our full data sets.

Each AVIRIS image was classified with each of the distance measures, so there were fifteen trials (5 images $\times$ 3 distance measures). Each trial used $K = 16$ clusters and stopped after convergence or 50 iterations. Results are shown in Table 1.

The total variance after classification for the Manhattan and Max distance measures was compared to the Euclidean distance measure. The total variance for both the Manhattan and Max distance measure trials was always higher than the Euclidean, by an amount that in one case was as high as thirty percent, but more typically was near ten percent.

Although the data are all integer-valued, and therefore so are the distances, the cluster centers need not be integer-valued, and for the results reported here, floating-point values were used. But we also performed some experiments with the C++ code comparing integer-valued centers and centers with an extra eight bits of precision. This was achieved by multiplying the data values by 256, and then continuing to work in the integer domain. We found that the fractional bits in the cluster centers did not affect the total variance substantially for any of the distance measures. There was no noticeable difference in the classified images because of the fractional bits either. Future work will consider the effect of further truncating the precision both of the data and of the centers.

6. CONCLUSIONS

Using both artificial models and real data, we compared the use of alternative distance metrics for k-means clustering. Although the Euclidean metric is theoretically optimal in terms of the cluster quality, but it is also expensive to compute, particularly in a hardware implementation. We considered two alternative distance metrics (the Manhattan and the Max), both of which would be more amenable to hardware implementation. Both in the data-independent Monte-Carlo studies, and in the comparisons using hyperspectral image data, we found that the alternative distances produced lower qualitiy clusters than those based on Euclidean distance. For the AVIRIS data, no consistent preference for Manhattan over Max could be concluded, but the Manhattan did have a lower rate of misclassification, and a smaller relative variance increase, in the idealized Monte-Carlo experiments.

7. REFERENCES

[1] Airborne Visible Infrared Imaging Spectrometer, 1999. `http://makalu.jpl.nasa.gov/aviris.html`.

[2] Vittorio Castelli and Thomas M. Cover. On the exponential value of labeled samples. *Pattern Recognition Lett.*, 16:105–111, 1995.

[3] Pi-Fuei Hsieh and David Landgrebe. Statistics enhancement in hyperspectral data analysis using spectral-spatial labeling, the EM algorithm, and the leave-one-out covariance estimator. *Proc. SPIE*, 3438:183–190, 1999.

[4] Robert A. Schowengerdt. *Techniques for Image Processing and Classification in Remote Sensing*. Academic Press, Orlando, 1983.

Image	Within-class Variance			Relative increase	
	Euclidean	Manhattan	Max	Manhattan	Max
f960323t01p02_r04_sc01.c.img	3.7533e+06	5.1121e+06*	3.9328e+06	0.3620	0.0478
f970410t01p02_r02_sc02.c.img	3.7806e+06	4.1296e+06	4.3915e+06	0.0923	0.1616
f970620t01p02_r03_sc02.c.img	1.0339e+07	1.1329e+07*	1.1089e+07	0.0958	0.0725
f970701t01p02_r07_sc01.c.img	9.4839e+06	9.9656e+06*	1.0726e+07*	0.0508	0.1309
f970801t01p02_r01_sc01.c.img	3.5210e+06	3.5597e+06	3.5828e+06*	0.0110	0.0175

Table 1: Comparions of within-class variance for clusterings of AVIRIS data using different variants of k-means. $K = 16$ clusters were used, and the algorithm was run for 50 iterations. An asterisk indicates that convergence was achieved before 50 iterations.

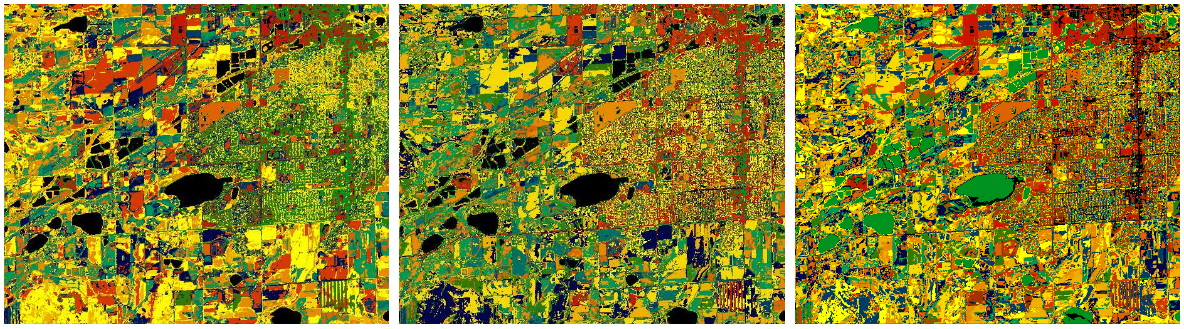

Figure 5: Clustering of AVIRIS image f9707 (Denver, Colorado), using (from left to right) Euclidean, Manhattan, and Max distance metrics. For this case, the Euclidean clustering produced the smallest within-class variance, followed by the Manhattan and then the Max.