Usage of dissimilarity measures and multidimensional scaling for large scale solar data analysis

Juan M. Banda[1], Rafal Anrgyk[2]

ABSTRACT: This work describes the application of several dissimilarity measures combined with multidimensional scaling for large scale solar data analysis. Using the first solar domain-specific benchmark data set that contains multiple types of phenomena, we investigated combinations of different image parameters with different dissimilarity measures in order to determine which combinations will allow us to differentiate our solar data within each class and versus the rest of the classes. In this work we also address the issue of reducing dimensionality by applying multidimensional scaling to our dissimilarity matrices produced by the previously mentioned combinations. By applying multidimensional scaling we can investigate how many resulting components are needed in order to maintain a good representation of our data (in a artificial dimensional space) and how many can be discarded in order to economize our storage costs. We present a comparative analysis between different classifiers in order to determine the amount of dimensionality reduction that can be achieve with said combination of image parameters, similarity measure and multidimensional scaling.

1. INTRODUCTION

In this work, we present the some of our steps toward the ambitious goal of building a Content Based Image Retrieval (CBIR) system for the Solar Dynamics Observatory (SDO) mission [24]. Our motivation for this work developed from the fact that with the large amounts of data that the SDO mission will be transmitting, hand labeling of these images will be an impossible task. There have been several successful CBIR systems for medical images [8] as well as in other domains [7]; none of them, however, have dealt with the volume of data that the SDO mission will generate.

After having investigated supervised and unsupervised attribute evaluation methods [1] that let us select the image parameters, which are the most relevant for our solar images. We are now confronted with the problem of determining the most informative dissimilarity measures for our benchmark dataset images and future images, since most classes and images are very similar to each other. Having this in mind we proceeded to experiment with twelve similarity measures that are widely used for images [10, 13, 19] in order to determine which ones would provide a better differentiation between our classes. In order to determine which combination of image parameters and similarity measures work best we created over 120 combinations, this will allow us to observe the behavior of all these combinations and help identify the most (and least) informative and useful.

Besides determining which combination of dissimilarity measure and image parameters works best, we also performed multidimensional scaling (MDS) to the resulting dissimilarity matrices. This method for visualization and dimensionality reduction has been widely used by researchers in different areas for image processing and retrieval [3, 4, 7, 21]. By applying MDS to our dissimilarity matrices, we want to achieve two things: 1) Have a 2D or 3D visualization of our image dataset dissimilarities that shows the class separation in a convenient way. 2) Verify the amount of dimensionality reduction that we can achieve with our data points mapped into a new artificial dimensional space.

In order to measure the degree of dimensionality reduction we can achieve, we set up two different ways of limiting the MDS components. We evaluate our work using comparative

[1] Montana State University, Bozeman, MT, juan.banda@cs.montana.edu
[2] Montana State University, Bozeman, MT, anrgyk@cs.montana.edu

analysis, where we compare the two different component selection methods by presenting comparative classification results for four different classifiers. This will allow us to determine how to select our components in order to achieve similar or even better classification results than with our original data. This dimensionality reduction is very important in terms of allowing us to considerably reduce our storage costs.

Our goal of publishing this work is not only to contribute to the existing knowledge on solar data analysis [1, 2, 31], but also to obtain valuable feedback from the community, especially from astrophysicists using image parameters different than the ones presented in our work. We are looking forward to build new collaborations with domain experts that are working on identifying individual solar phenomena and proceed with additions to our previously published benchmark data set and the CBIR system in order to better serve its purpose. Since the SDO mission has recently launched, the need to accurately detect and classify different types of solar phenomena in an automated way becomes of vital importance. We are open for discussion, and would greatly appreciate any feedback.

With the foundation framework presented here, other astrophysicists can greatly benefit from knowing which image parameters/distance measure combinations work well and could improve their work on classification of specific solar phenomenon. As we noted on [1], the results are very domain and individual solar phenomena specific allowing researchers working on a particular type of solar events (i.e. flares) to use the combination of image parameters/distance measures that better serve their classification purposes.

The rest of the paper is organized in the following way: a background is presented in Sec. 2. In Sec. 3 we present our experiments and the results produced. Sec. 4 presents the overall conclusions reached based on the experiment results. Sec. 5 includes the future work.

2. BACKGROUND

2.1 Benchmark dataset

Our dataset was first introduced in [1] consists of 1,600 images divided in 8 equally balanced classes representing 8 types of different solar phenomena. All of our images are 1,024 by 1,024 pixels.

Table 1. Characteristics of our benchmark data set

Event Name	# of images retrieved	Wavelength
Active Region	200	1600
Coronal Jet	200	171
Emerging Flux	200	1600
Filament	200	171
Filament Activation	200	171
Filament Eruption	200	171
Flare	200	171
Oscillation	200	171

The benchmark data set both in its original and pre-processed format is freely available to the public via Montana State University's server [27]. Because of promising results obtained during our preliminary investigations [2] and some earlier works [14], we choose to segment our images using an 8 by 8 grid for our image parameter extraction and labeling.

In this work, each image was transformed into ten 64-bin histograms, each bin representing the value of the each image parameter (table 2) extracted for each grid cell. We chose to treat each image parameter separately since want to determine their usefulness and behavior with the different dissimilarity measures.

2.2 Image parameters

Based on our literature review, we decided that we would use some of the most popular image parameters used in different fields such as medical images, text recognition, natural scene images and traffic images [5, 6, 8, 9, 11, 20, 29]. Since the usefulness of all these image parameters has shown to be very domain dependent, we performed our own investigation on the evaluation of this image parameters, which was published in [1].

The ten image parameters that we used for this work are presented on table 2. In our earlier work, we started with a larger list of parameters but we have been discarding them based on computational expense, performance and relevance [1, 2]. Please note that these image parameters are not exhaustive and there are a very large number of other parameters that we could have tested.

Table 2. List Of Extracted Image Parameters

Label	Image parameter
P1	Entropy
P2	Fractal Dimension
P3	Mean
P4	3rd Moment (skewness)
P5	4th Moment (kurtosis)
P6	Relative Smoothness
P7	Standard Deviation
P8	Tamura Contrast
P9	Tamura Directionality
P10	Uniformity

2.3 Dissimilarity measures

We selected twelve dissimilarity measures to use for comparison purposes. Based on our literature review, we believe that the measures selected are widely used in image analysis and produce good results when applied to images in other domains [10, 13, 19]. Since we work on very similar image data we decided to investigate different measures in order to verify how well they differentiate our images between our solar phenomena classes and mark similarities within the classes themselves. We will address this later in our experiment section, where we present plots of dissimilarity matrices.

For the first eight measures given an *m*-by-*n* data matrix X (in our case it contains m=1600 histograms and n=64 bins), which is treated as *m* (1-by-*n*) row vectors x_1, x_2, ..., x_m, the various distances between the vector x_s and x_t are defined as follows:

1) ***Euclidean distance [30]:*** Defined as the distance between two points give by the Pythagorean Theorem. Special case of the Minkowski metric where p=2.

$$D_{st} = \sqrt{(x_s - x_t)(x_s - x_t)'} \quad (1)$$

2) ***Standardized Euclidean distance [30]:*** Defined as the Euclidean distance calculated on standardized data, in this case standardized by the standard deviations.

$$D_{st} = \sqrt{(x_s - x_t)V^{-1}(x_s - x_t)'} \quad (2)$$

Where V is the n-by-n diagonal matrix whose j^{th} diagonal element is $S(j)^2$, where S is the vector of standard deviations.

3) ***Mahalanobis distance [30]:*** Defined as the Euclidean distance normalized based on a covariance matrix to make the distance metric scale-invariant.

$$D_{st} = \sqrt{(x_s - x_t)C^{-1}(x_s - x_t)'} \quad (3)$$

Where C is the covariance matrix

4) ***City block distance [30]:*** Also known as Manhattan distance, it represents distance between points in a grid by examining the absolute differences between coordinates of a pair of objects. Special case of the Minkowski metric where p=1.

$$D_{st} = \sum_{j=1}^{n} \left| x_{sj} - x_{tj} \right| \quad (4)$$

5) ***Chebychev distance [30]:*** Measures distance assuming only the most significant dimension is relevant. Special case of the Minkowski metric where p = ∞.

$$D_{st} = \max_j \left\{ \left| x_{sj} - x_{tj} \right| \right\} \quad (5)$$

6) ***Cosine distance [26]:*** Measures the dissimilarity between two vectors by finding the cosine of the angle between them.

$$D_{st} = 1 - \frac{x_s x_t'}{\sqrt{(x_s x_s')(x_t x_t')}} \quad (6)$$

7) ***Correlation distance [26]:*** Measures the dissimilarity of the sample correlation between points as sequences of values.

$$D_{st} = 1 - \frac{(x_s - \bar{x}_s)(x_t - \bar{x}_t)'}{\sqrt{(x_s - \bar{x}_s)(x_s - \bar{x}_s)'}\sqrt{(x_t - \bar{x}_t)(x_t - \bar{x}_t)'}} \quad (7)$$

Where $\bar{x}_s = \frac{1}{n}\sum_{j=1}^{n} x_{sj}$ and $\bar{x}_t = \frac{1}{n}\sum_{j=1}^{n} x_{tj}$

8) ***Spearman distance [25]:*** Measures the dissimilarity of the sample's Spearman rank [25] correlation between observations as sequences of values.

$$D_{st} = 1 - \frac{(r_s - \bar{r}_s)(r_t - \bar{r}_t)'}{\sqrt{(r_s - \bar{r}_s)(r_s - \bar{r}_s)'}\sqrt{(r_t - \bar{r}_t)(r_t - \bar{r}_t)'}} \quad (8)$$

Where r_{sj} is the rank of x_{sj} taken over $x_{1j}, x_{2j}, \ldots x_{mj}$, r_s and r_t are the coordinate-wise rank vectors of x_s and x_t, i.e., $r_s = (r_{s1}, r_{s2}, \ldots r_{sn})$ and $\bar{r}_s = \frac{1}{n}\sum_{j=1}^{n} r_{sj} = \frac{(n+1)}{2}$, $\bar{r}_t = \frac{1}{n}\sum_{j=1}^{n} r_{tj} = \frac{(n+1)}{2}$

Since our focus is on comparing image histograms, we present the next for measures in terms of histograms.

9) ***Hausdorff Distance [17]:*** Intuitively defined as the maximum distance of a histogram to the nearest point in the other histogram.

$$DH(H,H') = \max\{\sup_{x\in H}\inf_{y\in H'} d(x,y), \sup_{y\in H'}\inf_{x\in H} d(x,y)\} \quad (10)$$

Where sup represents the supremum, inf the infimum, and d(x,y) represents any distance measure between two points, in our case we used Euclidean distance.

10) ***Jensen–Shannon divergence (JSD) [15]:*** Also known as total divergence to the average, Jensen–Shannon divergence is a symmetrized and smoothed version of the *Kullback–Leibler divergence*.

$$JD(H,H') = \sum_{m=1}^{n} H_m \log\frac{2H_m}{H_m + H'_m} + H'_m \log\frac{2H'_m}{H'_m + H_m} \quad (11)$$

11) χ^2 ***distance [22]:*** Measures the likeliness of one histogram being drawn from another one.

$$\chi^2(H,H') = \sum_{m=1}^{n} \frac{H_m - H'_m}{H_m + H'_m} \quad (12)$$

12) ***Kullback–Leibler divergence (KLD) [12]***: Measures the difference between two histograms H and H'. Often intuited as a distance metric, the KL divergence is not a true metric since the KL divergence from H to H' is not necessarily the same as the KL divergence from H' to H.

$$KL(H,H') = \sum_{m=1}^{n} H_m \log\frac{H_m}{H_m} \quad (13)$$

Since this is the only non-symmetric measure we used for this work. We treated it as a directed measure and considered H-H' and H'-H as two different distances.

2.4 Multidimensional scaling and curve fitting

Multidimensional scaling (MDS) is a set of statistical techniques used for the exploration of similarities or dissimilarities in data, in the field of Information visualization. MDS is also commonly used as a method for dimensionality reduction for large similarity or dissimilarity matrices [3, 4, 7, 21]. We used the classical multidimensional scaling approach since we have input matrices giving dissimilarities between pairs of items (produced by our similarity measures). This process will output a coordinate matrix whose configuration minimizes a loss function called *strain*.

With our resulting MDS matrices we have a new dimensional space, where each component of the matrix determines how relevant they are in discerning similarities within the original data (similar to PCA or SVD). However, one of the main issues behind MDS is that does not provide an explicit mapping function governing the relationship between patterns in the input space and in the projected space [18].

Based on the magnitudes of each of the resulting MDS components, we decided to use exponential curve fitting in order to be able to threshold the optimal number of components needed in order to reduce dimensionality and still retain valuable components in order to produce good classification results. For comparative purposes we also opted for a far simpler approach of only selecting 10 components and discarding the rest, this would allow us to verify how much will a few (sometimes many) extra components will increase or decrease our classification results.

2.5 Classifiers and boosting algorithms

We selected Naïve Bayes and Support Vector Machines (SVM) with a linear kernel function as our linear classifiers and C4.5 as a decision tree classifier. Linear classifiers achieve the grouping of items that have similar feature values into groups by making a classification decision based on the value of the linear combination of the features. Whereas C4.5 uses entropy-based information gain measure to split samples into classes.

Based on the dimensionality and distribution of the values from our image parameters, we decided to investigate the results of a decision tree classifier in addition to the linear classifiers. A decision tree classifier has the goal of creating a model that predicts the value of a target variable based on several input variables. Each interior node corresponds to one of the input variables; there are edges to the children for each of the possible values of that input variable. Each leaf represents a value of the target variable given the values of the input variables represented by the path from the root to the leaf.

We selected Adaptive Boosting as our boosting algorithm in order to determine the effectiveness of boosting on our data. AdaBoost is adaptive in the sense that subsequent classifiers built are tweaked in favor of those instances that were misclassified by previous iterations of the classifier. This algorithm is sensitive to noisy data and outliers. But it is less susceptible to the overfitting problem than affects most learning algorithms.

Note that we use these classifiers in order to present a comparative analysis of our experiments. In this paper we are not trying to find the best classification results or tweak the classifiers to perform at its best. We are trying to determine how many components of our new artificial data space we can be omitted without significant decrease in classification results.

3. APPROACH AND EXPERIMENTS

All experiments were performed using Matlab R2009b. For the exponential curve fitting we used the Ezyfit Tool box [16]. The classification experiments were performed using

WEKA 3.6.1. These programs were run on a PC with AMD Athlon II X4 2.60 Ghz Quad Core processor with 8 GB's of RAM and Windows XP 64-bit Edition.

3.1 Dissimilarity matrix calculations

In order to correctly evaluate each of the extracted image parameters (table 2) we need to treat them individually. We created a 64 bin histogram (from our 8x8 grid segmentation) per image parameter, per image. In order to use these histograms correctly when calculating the KLD and JSD measures we need to make sure the sum of the bins adds to one. To achieve this, we normalized every single parameter per image in the following way:

$$NH_m = \frac{H_m}{\sum_{m=1}^{n} H_m} \tag{14}$$

Where n=64, since we have a total of 64 bins.

For each bin in the histograms, this allows us to scale our histograms and preserve their shape. For bins equaling zero, we had to add a very small quantity ($0.1x10^{-8}$) in order to avoid divisions by zero on the KLD measure.

After all our data has been normalized this way, we proceeded to calculate the pair wise distance between the histograms using Matlab's `pdist` function. As this function is highly optimized for performance, the computation time for our first 8 measures is very low. The Haussdorff, KLD, JSD and χ^2 distances where implemented from scratch and yield higher computational expense due to their nature.

In total we produced a total of 130 dissimilarity matrices (13 measures, counting KLD H-H' and H'-H, times a total of 10 different image parameters). All these dissimilarity matrices are symmetric, real and positive valued, and their diagonals are zero, fitting the classical multidimensional scaling requirements.

These dissimilarity matrices help us to identify which image parameters and measures provide nice differentiation for our images between the 8 different classes on our dataset. In this paper we will focus on three of the most informative parameter-measure combinations we generated (good and bad), but you can access all these matrices online at [28]. Here the classes of our benchmark are separated on the axes, every 200 units (images) the next class starts. The classes are ordered in the same way as on table 2.

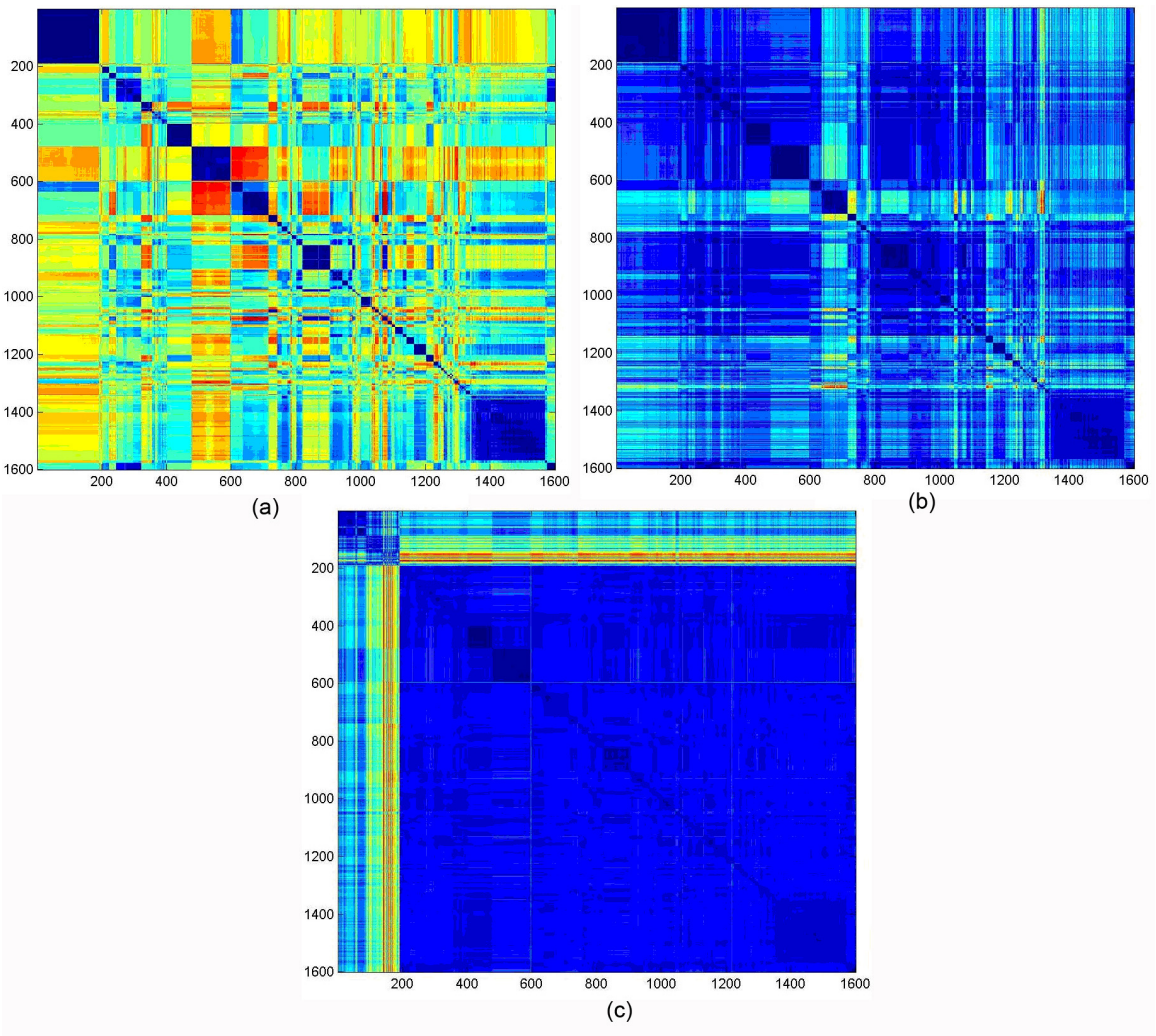

Figure 1. Scaled Image plot of dissimilarity matrix for (a) Correlation measure with image parameter mean, (b) JSD measure with image parameter mean, (c) Chebychev measure with image parameter Relative Smoothness

As we can see in figure 1(a), this combination of similarity measure (correlation) and image parameter (mean) produces a nice separation between classes. Blue means high similarity, and red means high dissimilarity. Figure 1(b) shows that the JSD measure produces and entirely different dissimilarity matrix for the same image parameter (mean) that highlights different similarities that the correlation measure reflected. Figure 1(c) is a clear example of a combination of similarity measure (Chebychev) and image parameter (relative smoothness) that highlights dissimilarities within only one class of the benchmark, but recognizes everything else are highly similar for the rest of the classes. This validates our idea of testing every image parameter individually, since there are combinations that will allow us to notice different relationships between measure/parameter that will allow us to differentiate images between classes.

The figure 2 presents the average time in minutes that is required to calculate **one** 1,600 by 1,600 dissimilarity matrix for each of the twelve dissimilarity measures. Note that the first 8 distances on average are very fast and optimized; this is due to the fact that we used Matlab's own pdist function to calculate them. The remaining 4 distances are our own

implementations and can be further optimized. We mention that KLD is times two since we need to consider H-H' and H'-H since the measure is not symmetric.

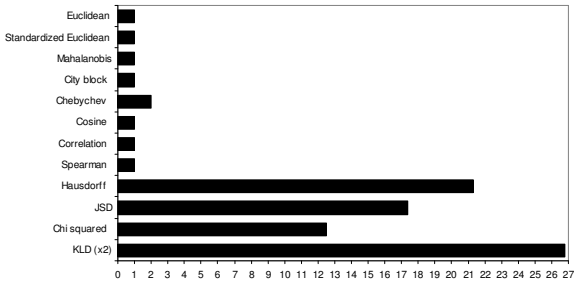

Figure 2. Average time in minutes that is required to calculate one 1,600 by 1,600 dissimilarity matrix

3.2 Multidimensional scaling and curve fitting

After generating our 130 dissimilarity matrices, we performed classical multidimensional scaling using Matlab's `cmdscale` function. MDS has been widely utilized in many image retrieval works to reduce dimensionality [3, 21], and to aid in the visualization of similar images in a convenient two and three dimensional plot [4]. However these works present results on a considerably smaller number of images and using a considerably smaller number of dimensions. The most commonly used MDS plots (maps) involve using the first two or three components of the outputted coordinate matrix.

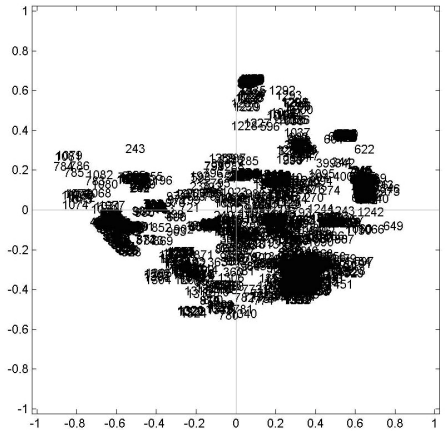

Figure 3. MDS map for the correlation measure with image parameter mean

As we suspected, on figure 3 we can't really identify a clear separation between our 8 different classes. We theorize that since our images themselves have high similarity we need a considerable amount of components in order to start to see separation between them. All the 130 MDS maps are available at [28], where we present an extended version of this paper as well as all the results of the experiments performed for in this work.

Like we mentioned before, MDS is also used for dimensionality reduction and we analyzed the magnitudes (importance) of the components in order to determine how many components we really need to maintain a good representation of our data in the new dimensional space, and how many components we can discard in order for us to reduce our storage costs.

In order to determine this number of components we plotted the magnitudes of each component. Since the MDS coordinate matrix output is ordered by importance, the

magnitudes should be decreasing as the number of component increases. In order for us to threshold this data we utilized exponential curve fitting [23] to find a function that would model this behavior and we could use to threshold the number of components needed. We utilized a 135 degree angle of the tangent line to this function in order to determine where to threshold and discard the components that their magnitudes where not providing significant improvement over the previous one.

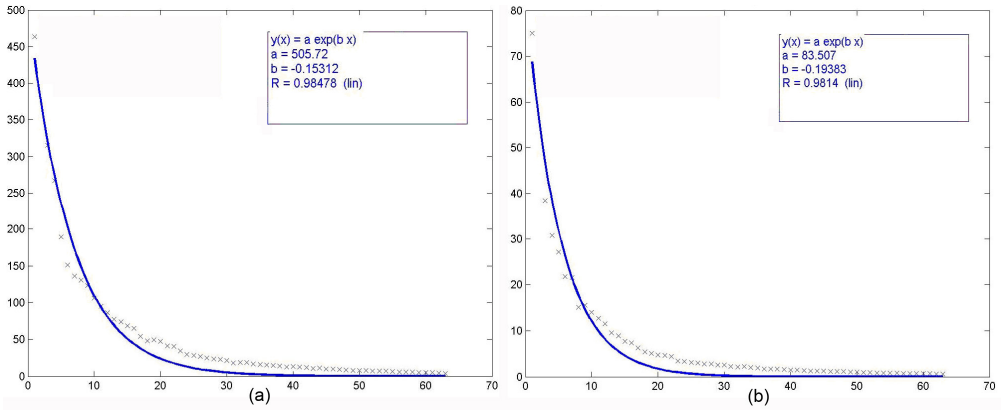

Figure 4. Exponential curve fitting for: (a) correlation measure with image parameter mean, (b) JSD measure with image parameter mean

As you can see from figure 4, we have the magnitudes of the components decreasing up to a certain point, and then the change is very minimal and thus not too important for the new dimensional space.

Based on these curve fitting results and the threshold output, we determined a specific number of components per combination of measure/image parameter. We can now determine how well this reduced dimensionality performs in our classification tasks on section 3.3.

3.3 Classification

Until now we have described how we applied the similarity measures to our image parameters and how MDS transformed them into a different dimensional space, one that will require, hopefully, less dimensions in order to represent our data in a similar way than originally. We now describe the classification experiments we performed on the resulting tangent thresholded components versus our original data. Since we noticed an empirical observation that after 10 components the decrease in their magnitudes stops being drastic (in most of the cases), we decided to take a somewhat naïve approach and perform a threshold of 10 components per similarity measure/image parameter combination of the same tangent thresholded components. All classification experiments were run using 10 fold cross-validation.

We ran a total of 270 different datasets through the 4 classifiers described in section 2.5. In the following plots we present the overall results of this classification experiments and after that we offer a more detailed explanation of the most interesting results.

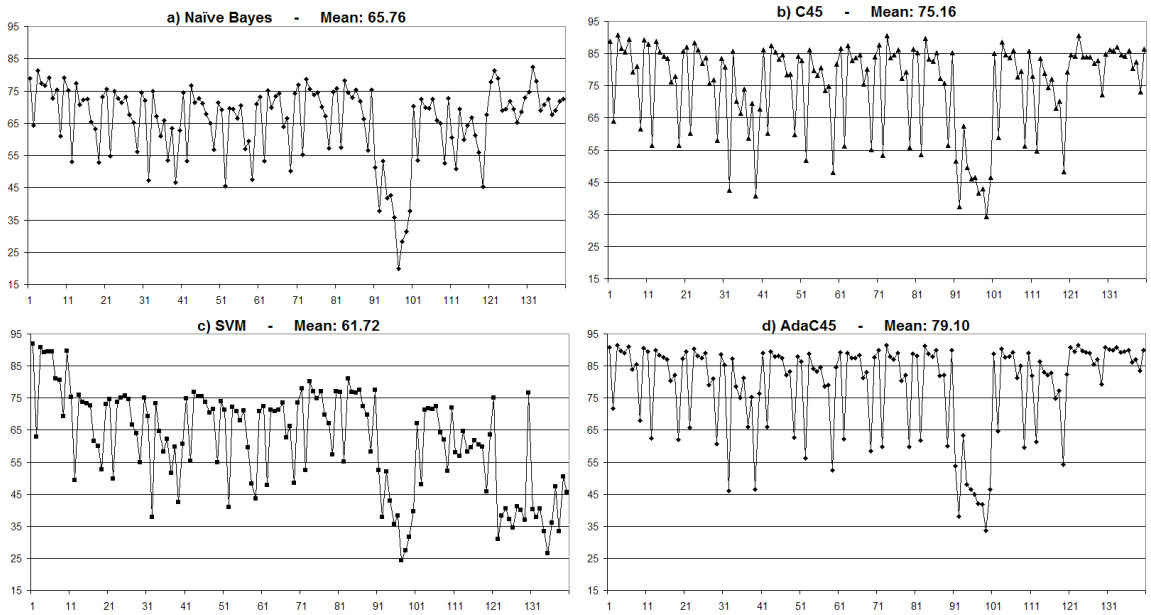

Figure 5. Percentage of correctly classified instances for the 10 component threshold

Figure 5 shows the classification accuracy of our selected classifiers on our 10 component per measure/image parameter combination. The first 10 columns indicate our original normalized dataset values with no measure or dimensionality reduction applied to them. The rest of the columns (11 to 140) indicate our measure/image parameter combinations in the order they are presented in section 2.3 and table 1. Individual charts presenting the classification results for each classifier are available at [28].

We can see from the figures that our 10 components only approach produces very similar classification results that our original data for most combinations of measure and image parameters. We can also notice the worst performing measure/image parameter combination is presented in columns 91 to 100 which correspond to the Hausdorff similarity measure. We will discus the rest of our general conclusions on the following section.

In figure 6 we present the resulting number of components to be used based on the tangent thresholding. The columns represent the 130 different image parameter/measure combinations with the omission of the first 10, which are the original dataset.

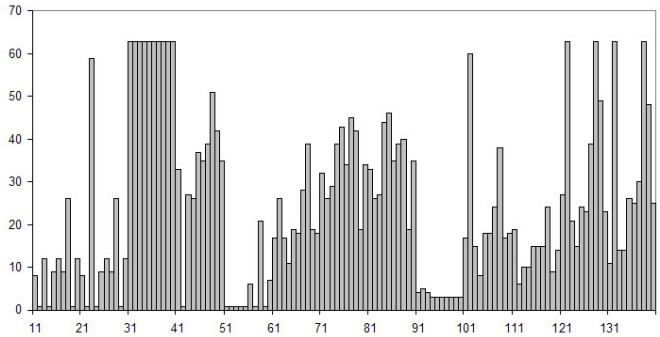
Figure 6. Number of components to use indicated by the tangent thresholding method.

In the next figure we present the tangent thresholded classification results. The number of components selected varied between 1 and 63 depending on the combination of

measure/image parameter. The columns are ordered the same way as in the previous figures.

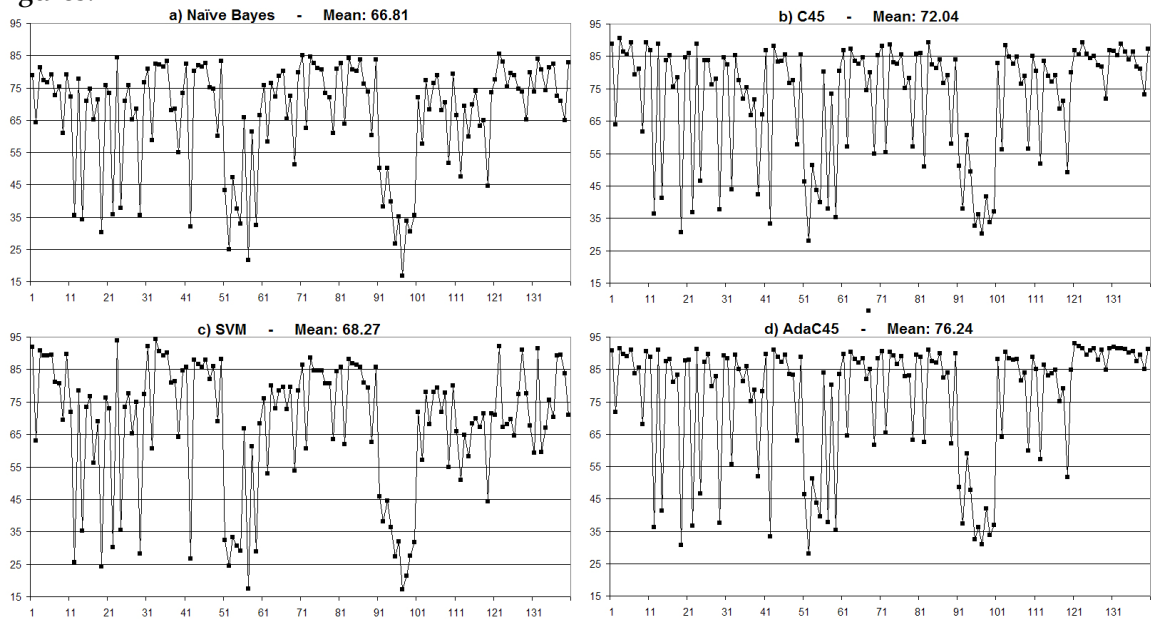

Figure 7. Percentage of correctly classified instances for the tangent-based component threshold

As we can observe in the thresholded components classification results, we get very similar results than with only 10 components, and in some cases we get considerable drops i.e., for the Chebychev measure (columns between 51-60), this is due to the fact that the thresholding selected less than 10 components per combination of measure/image parameter and in some instances even only 1 component. An interesting thing to notice is that the overall classification percentages increased consistently for the KLD H-H' and H'-H combinations, but also due to the fact that the thresholding selected 63 components for the several of image parameter combinations.

With the previously mentioned results for both of the tangent thresholding and the 10 component limiting we can observe that even with only 10 components we can achieve good accuracy results (around 80-90%) for the selected classifiers. We can also see which image parameters perform the best with which measures, one of our objectives with this paper.

The Support Vector machines classifier produces better results when it has a higher number of components, and achieves its best with the original data since it has the highest number of data points. For a better comparison, we decided to show only the results from the Naïve Bayes, C45 and Adaboost C45 classifiers, since they tend to not be influenced as much by the number of data points that they are using for classification.

In the next table we present the top 5 classification results for each the tangent thresholded and the 10 components limited datasets.

Table 3. Top 5 classification results for 10 component limited and tangent thresholded dimensionality reduction exp.

10 Component limit					
	Bayes		C45		AdaC45
Distance-KLD B-A-Feature-FracDim	82.44	Original Data-Mean	90.69	Original Data-Mean	91.56
Original Data-Mean	81.31	Distance-correlation-Feature-Mean	90.63	Distance-correlation-Feature-Mean	91.56
Distance-KLD A-B-Feature-FracDim	81.25	Distance-KLD A-B-Feature-Mean	90.63	Distance-KLD A-B-Feature-Mean	91.44
Original Data-Uniformity	79.19	Distance-spearman-Feature-Mean	89.63	Distance-spearman-Feature-Mean	91.13
Original Data-RelSm	79.00	Original Data-RelSm	89.38	Original Data-RelSm	91.00
Tangent Threshold					
	Bayes		C45		AdaC45
Comp-63-Distance-KLD A-B-Feature-FracDim	85.50	Original Data-Mean	90.69	Comp-27-Distance-KLD A-B-Feature-Entropy	92.94
Comp-32-Distance-correlation-Feature-Entropy	85.06	Original Data-RelSm	89.38	Comp-63-Distance-KLD A-B-Feature-FracDim	92.13
Comp-29-Distance-correlation-Feature-Mean	84.69	Comp-21-Distance-KLD A-B-Feature-Mean	89.38	Comp-11-Distance-KLD B-A-Feature-Entropy	91.88
Comp-59-Distance-seuclidean-Feature-Mean	84.31	Original Data-Uniformity	89.19	Original Data-Mean	91.56
Comp-27-Distance-spearman-Feature-Mean	84.19	Comp-27-Distance-spearman-Feature-Mean	89.19	Comp-14-Distance-KLD B-A-Feature-Mean	91.56

4. CONCLUSIONS

With the ambitious tasks of analyzing all the combinations between image parameters and dissimilarity measures, we managed to create a solid foundation of information that will allow us to determine what works best for the classification of different solar phenomena. The results of these experiments also allowed us to show that we can considerably reduce our dimensionality and still get good (and sometimes even better) classification results.

Some dissimilarity measures, like Correlation, Euclidean, KLD and JSD, allowed us to easily discern the dissimilarities between our images in our dataset and provided different levels of relevance between different image parameters. As every researcher knows, not everything works always, and with this work we can actually notice what works well and for when in terms of solar images.

While not all dissimilarity measures performed equally well, we now know which ones to remove and omit due to their computational expense for future experiments (i.e. Hausdorff measure).

In terms of dimensionality reduction, we managed to achieve very similar (and sometimes better) classification results than with the original data. The thresholding of these components provided good performance; improving sometimes the classification results of the limiting of 10 components, but with its added computational expense the improvements where not considerable. For future work we will utilize this limiting of 10 components with the certainty that in our domain specific task, the thresholding did not provide considerable improvements. Astrophysicists using a similar machine learning approach to classify individual phenomena can be benefited by our approach on how to select the number of components and might choose to implement it in order to reduce their storage costs and possibly speed up their retrieval times.

With the massive amounts of experiments performed, in this medium we lack the proper space to display all the results we produced. All the dissimilarity matrices, MDS maps, exponential curve fitting plots, and all the classification results are presented on [28] for researchers interested in all these results. We also included all the Matlab and WEKA files produced in order for people to easily replicate these results.

5. FUTURE WORK

With all the different dissimilarity measures and image parameters in the community, we would greatly appreciate any feedback from other researchers using different measures/parameters to the ones presented in this paper and expand our research.

We are currently working with different dimensionality reduction methods other than MDS, such as Principal Component Analysis and Singular Value Decomposition among others. These two methods have the advantage of producing mapping functions in order to transform new data into the artificial dimensional space created by them. This will allow us to use a particular training dataset and a new test dataset in order to create more accurate classification predictions.

As we mentioned before, all the classifiers used in this paper, were created using the default WEKA settings for them. The classification results are for comparative purposes and in no way they reflect the results that can be obtained after fine tuning the settings of these classifiers. We are currently working on this issue, and we expect to publish soon results of fine-tuned classifiers in a future paper. We also expect to add the number of classifiers used to have a more comprehensive evaluation of them in the future.

Lastly, we continue working towards the goal of creating a fully working CBIR system for the SDO mission, and with this work as well as our previous papers, we are getting closer to this ambitious goal.

6. REFERENCES

[1] J. Banda and R. Anrgyk "An Experimental Evaluation of Popular Image Parameters for Monochromatic Solar Image Categorization". FLAIRS-23: Proceedings of the twenty-third international Florida Artificial Intelligence Research Society conference, Daytona Beach, Florida, USA, May 19–21 2010 (to be published). 2010.

[2] J. Banda and R. Anrgyk "On the effectiveness of fuzzy clustering as a data discretization technique for large-scale classification of solar images." In Proceedings of the 18th IEEE International Conference on Fuzzy Systems (Jeju Island, Korea, August 2009):2019-2024, 2009.

[3] M. Beatty, B. S. Manjunath, "Dimensionality Reduction Using Multi-Dimensional Scaling for Content-Based Retrieval," Image Processing, International Conference on, International Conference on Image Processing (ICIP'97) - Volume 2: 835, 1997.

[4] I. Borg, P.J.F Groenen, Modern multidimensional scaling: Theory and applications, Springer Verlag :191-193, 1997.

[5] E. Cernadas, P. Carriön P., P. Rodriguez, E. Muriel, and T. Antequera. "Analyzing magnetic resonance images of Iberian pork loin to predict its sensorial characteristics". Comput. Vis. Image Underst. Vol. 98 (2):345-361. 2005.

[6] B.B Chaudhuri, S. Nirupam. "Texture Segmentation Using Fractal Dimension." IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 17 (1): 72-77, 1995.

[7] R. Datta, J Li and Z. Wang. "Content-based Image Retrieval – Approaches and Trends of the New Age". In ACM Intl. Workshop on Multimedia Information Retrieval, ACM Multimedia. 2005.

[8] T. Deselaers, D. Keysers, and H. Ney. "Features for Image Retrieval: An Experimental Comparison" Information Retrieval, vol. 11, issue 2, Springer (The Netherlands 03/2008) :77-107. 2008.

[9] V. Devendran, T. Hemalatha, W. Amitabh. "SVM Based Hybrid Moment Features for Natural Scene Categorization". International Conference on Computational Science and En-gineering Vol. 1: 356-361. 2009.

[10] G. D Guo, A.K. Jain, W.Y Ma, H.J Zhang,et. all, "Learning similarity measure for natural image retrieval with relevance feedback". IEEE Transactions on Neural Networks. Volume 13 (4): 811--820, 2002

[11] S.S Holalu and K. Arumugam. “Breast Tissue Classification Using Statistical Feature Ex-traction Of Mammograms.” Medical Imaging and Information Sciences, Vol. 23 (3):105-107, 2006.

[12] S. Kullback, R.A. Leibler "On Information and Sufficiency". Annals of Mathematical Statistics 22 (1): 79–86. 1951.

[13] R. Lam, H. Ip, K. Cheung, L. Tang, R. Hanka, "Similarity Measures for Histological Image Retrieval," 15th International Conference on Pattern Recognition (ICPR'00) - Volume 2: 2295. 2000.

[14] R. Lamb, “An Information Retrieval System For Images From The Trace Satellite,” M.S. thesis, Dept. Comp. Sci., Montana State Univ., Bozeman, MT. 2008.

[15] J. Lin. "Divergence measures based on the shannon entropy". IEEE Transactions on Information Theory 37 (1): 145–151. 2001.

[16] F. Moisy “EzyFit 2.30” [Online], Available: http://www.mathworks.com/matlabcentral/fileexchange/10176 [Accessed: May 12, 2010]

[17] J. Munkres. Topology (2nd edition). Prentice Hall, 1999. pp 280-281.

[18] A. Naud “Neural and Statistical Methods for the Visualization of Multidimensional Data” Ph.D Thesis Uniwersytet Mikołaja Kopernika w Toruniu. P 84-85, 2001.

[19] T. Ojala, M. Pietikainen, and D. Harwood. A comparative study of texture measures with classification based feature distributions. Pattern Recognition, 29(1):51–59. 1996.

[20] A.P Pentland “Fractal-based description of natural scenes.” IEEE Trans. on Pattern Analysis and Machine Intelligence Vol. 6:661-674, 1984.

[21] Y. Rubner, L.J Guibas, and C. Tomasi, C. “The earth mover’s distance, multi-dimensional scaling, and color-based image retrieval” Proceedings of the ARPA Image Understanding Workshop :661—668, 1997

[22] A. Shahrokni. “Texture Boundary Detection for Real-Time Tracking”
Computer Vision - ECCV 2004: 566-577. 2004.

[23] R.N Shepard, “Multidimensional scaling, tree-fitting, and clustering” Science Vol. 210 (4468): 390—398, 1980.

[24]Solar Dynamics Observatory [Online], Available: http://sdo.gsfc.nasa.gov/. [Accessed: May 12, 2010]

[25] C. Spearman, "The proof and measurement of association between two things" Amer. J. Psychol. ,V 15 :72–101. 1904

[26] P.-N. Tan, M. Steinbach & V. Kumar, "Introduction to Data Mining", Addison-Wesley 500, 2005.

[27] TRACE Dataset (MSU) [Online], Available: http://www.cs.montana.edu/angryk/SDO/data/ [Accessed: May 12, 2010]

[28] Usage of dissimilarity measures and multidimensional scaling for large scale solar data analysis Website [Online], Available: http://www.jmbanda.com/CIDU2010/ [Accessed: May 19, 2010]

[29] C. Wen-lun, S. Zhong-ke, F. Jian. “Traffic Image Classification Method Based on Fractal Dimension.” IEEE International Conference on Cognitive Informatics Vol. 2: 903-907, 2006.

[30] K. Yang, J. Trewn. Multivariate Statistical Methods in Quality Management. McGraw-Hill Professional; February 24, 2004 pp. 183-185.

[31] V. Zharkova, S. Ipson, A. Benkhalil, and S. Zharkov. “Feature recognition in solar images.” Artificial Intelligence Review, Vol. 23(3):209–266. 2005.