Quantitative comparison of linear and non-linear dimensionality reduction techniques for solar image archives

Juan M. Banda[1], Rafal A. Angryk[2], Petrus C. Martens[3]

Department of Computer Science [1,2], Department of Physics [3]
Montana State University,
Bozeman, MT, 59715 USA.
{juan.banda, angryk}@cs.montana.edu [1,2], martens@physics.montana.edu [3]

Abstract

This work investigates the applicability of several dimensionality reduction techniques for large scale solar data analysis. Using the first solar domain-specific benchmark dataset that contains images of multiple types of phenomena, we investigate linear and non-linear dimensionality reduction methods in order to reduce our storage costs and maintain an accurate representation of our data in a new vector space. We present a comparative analysis between several dimensionality reduction methods and different numbers of target dimensions by utilizing different classifiers in order to determine the percentage of dimensionality reduction that can be achieved on solar data with said methods, and to discover the method that is the most effective for solar images.

Introduction

In this work, we present our dimensionality reduction analysis aimed towards the ambitious goal of building a large-scale Content Based Image Retrieval (CBIR) system for the Solar Dynamics Observatory (SDO) mission [1]. Our motivation for this work comes from the fact that with the large amounts of data that the SDO mission started transmitting, hand labeling (commonly used by solar physicist in the last decades) of these images is simply impossible. There have been several successful CBIR systems for medical images [2] as well as in other domains [3]; none of them, however, have dealt with the volume of data that the SDO mission generates. This NASA mission, only with its Atmospheric Imaging Assembly (AIA), generates eight 4096 pixels x 4096 pixels images every 10 seconds. This leads to a data transmission rate of approximately 700 gigabytes per day only from the AIA component (the entire mission is expected to be sending about 1.5 terabytes of data per day, for a minimum of 5 years).

With such a massive pipeline choosing redundant dimensions on our data will lead to unnecessary data storage, and high search and retrieval costs in our repository. Based on these complications, one of the main goals of this work is to determine the percentage of dimensionality reduction we can achieve using the best methods while maintaining a high quality parameter-based representation of the solar images.

Dimensionality reduction methods have been shown to produce accurate representations of high dimensional data in a lower dimensional space with very domain specific results in other image retrieval application domains [4-8]. In this work we investigate four linear and four non-linear dimensionality reduction methods with eight different numbers of target dimensions as parameters for each, in order to present a comparative analysis.

The novelty of our work is to determine which dimensionality reduction methods produce the best and most consistent results and with which classifiers, on our specific image parameters selected for solar data [17]. Due to domain-specific results, reported by multiple-researchers working on dimensionality reduction in the past [4-8], we believe our results will be of special interest to researchers from the field of medical image analysis, as these images seem to be the closest to our dataset [21. We also identify some interesting combinations of dimensionality reduction methods and classifiers that behave differently across the presented datasets. Our research problem in Solar physics is of great practical relevance for Earth's climate since solar flares endanger the lives of passengers on commercial airline routes going over the poles, interrupt radio communications in bands the military uses, can (and have) knocked down power grids. The systematic feature recognition and the study of the metadata, is a key component of the ultimate prediction of solar activity (space weather).

The rest of the paper is organized in the following way: A background overview is presented in the following section. After that we present an overview of the steps and experiments we performed together with our observations. The last includes our conclusions and the future work we propose to complete in order to prepare all parts of a solar CBIR system for integration.

Background

Most of the current works in solar physics focus on individual types of solar phenomena. Automatic identification of flares, on the SDO mission, is performed by an algorithm created by Christe et al. [10] which works well for noisy and background-affected light curves. This approach will allow detection of simultaneous flares in different active regions. Filament detection for the SDO mission will be provided by the “Advanced Automated

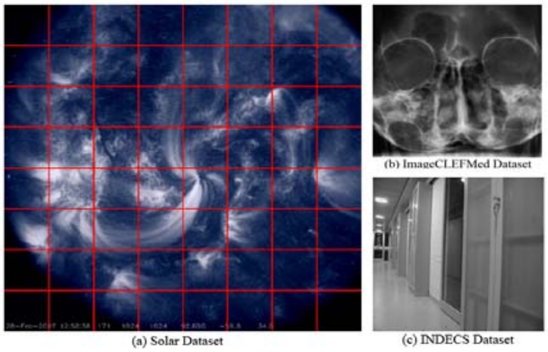
Filament Detection and Characterization Code" [11]. As for the coronal jet detection and parameter determination algorithms, these SDO methods are described in detail in [12]. In order to detect active regions, the SDO pipeline will use the Spatial Possibilistic Clustering Algorithm (SPoCA). Not until recently Lamb et al. [15] discussed creating an example based Image Retrieval System for the TRACE repository. This is the only attempt, that we are aware of, that involves trying to find a variety of phenomena, with expectation of building a large-scale CBIR system for solar physicists.

Some comparisons between dimensionality reduction methods for image retrieval have been performed in the past [4-8], these works constantly encounter the fact that results are very domain-specific and that performance of the non-linear versus linear dimensionality reduction methods has been shown to be dependent of the nature of the dataset (natural vs. artificial) [16]. We expect to find interesting properties of our dataset with the application of different types of dimensionality reduction methods.

Benchmark Datasets

The dataset, first introduced in [17], consists of 1,600 images divided in 8 equally balanced classes representing 8 types of different solar phenomena (each having 200 images). All of our images are grayscale and 1,024 by 1,024 pixels. The solar phenomenons included in the dataset are: Active Region, Coronal Jet, Emerging Flux, Filament, Filament Activation, Filament Eruption, Flare and Oscillation.

The benchmark dataset both in its original and pre-processed format is freely available to the public via Montana State University's server [18]. Because of promising results obtained during earlier investigations [15, 19], we choose to segment our images using an 8 by 8 grid for our image parameter extraction (as seen on fig. 1).


(a) Solar Dataset

(b) ImageCLEFMed Dataset

(c) INDECS Dataset

Figure 1. Grid based segmentation applied to our solar dataset prior to parameter extraction (a). (b) and (c) show samples of images of the other datasets tested.

Based on the 8 by 8 grid segmentation and our ten image parameters per each cell, our current benchmark dataset has 640 dimensions per image. Since the SDO images are 4 times bigger, they will produce a total of 10,240 dimensions per image and at a cost of 540 kilobytes per dimension (per day) this will get very expensive to store and search.

For comparative purposes we utilized subsets (matching the quantity of 1,600 images in eight equally balanced classes) of the following datasets: INDECS [44] and ImageCLEFmed [43] 2005. The INDECS dataset provides office images that are very dissimilar from each other and from our own type of images. On the other hand, the ImageCLEFmed [43] 2005 dataset provides several classes of medical images that somehow resemble ours due to the fact that both images are greyscale and feature fuzzy objects with just a few well defined persistent visual characteristics.

Image parameters (a.k.a Image Features)

Based on our literature review, we decided that we would use some of the most popular image parameters (as called in the field of Solar physics, but referred to as image features in computer vision areas) used in different fields such as medical images, text recognition, natural scene images and traffic images [2, 13, 20-22]. Since the usefulness of all these image parameters has shown to be very domain dependent, we selected these parameters, based on the evaluation published in [11, 17 and 19] that covered both supervised and unsupervised attribute evaluation methods and techniques.

The ten image parameters that we used for this work are: Entropy, Fractal dimension, Mean, Skewness, Kurtosis, Relative Smoothness, Standard Deviation, Tamura Contrast, Tamura Directionality and Uniformity.

Dimensionality Reduction Methods

Based on our literature review, we decided to utilize four different linear and four non-linear dimensionality reduction methods. As shown by others [5, 8, 16] linear dimensionality reduction methods have proved to perform better than non-linear methods in most artificial datasets and some natural datasets. However all these results have been very domain dependent. Classical methods like PCA and SVD are widely used as benchmarks to provide a comparison versus the newer non-linear methods. We selected eight different methods based on (1) their popularity in the literature, (2) the availability of a mapping function or method to map new unseen data points into the new dimensional space, (3) computational expense, and (4) the particular properties of some methods such as the preservation of local properties between the data and the type of distances between the data points (e.g. Euclidean versus geodesic).

Linear dimensionality reduction methods

Principal Component Analysis (PCA) [23]

PCA is defined as an orthogonal linear transformation that transforms data into an artificial space such that the greatest variance by any projection of the data comes to lie on the first principal component, and so on.

For a data matrix, M^T, where each row represents a different observation on the dataset, and each column contains the variables, the PCA transformation is given by:

$$Y^T = M^T U = V \sum^T \qquad (1)$$

where the matrix Σ is an m-by-n diagonal matrix with nonnegative real numbers on the diagonal and $U\Sigma\ V^T$ is the singular value decomposition of M.

Singular Value Decomposition (SVD) [24]

Defined for a m-by-n matrix M as the factorization of the form

$$M = U \sum V^T \qquad (2)$$

where U is an m-by-m unitary matrix, the matrix Σ is m-by-n diagonal matrix with nonnegative real numbers on the diagonal, and V^T denotes the conjugate transpose of V, an n-by-n unitary matrix.

Normally, the diagonal entries $\sum_{i,i}$ are ordered in a descending way. The columns of V form a set of orthonormal input basis vector directions for M. (The eigenvectors of $M^T M$.) The columns of U form a set of orthonormal output basis vector directions for M. (The eigenvectors of MM^T).

The main difference, between PCA and SVD is that SVD is directly applied to an m-by-n matrix (m-by-n long feature vectors), while in PCA is applied to a covariance matrix (n-by-n).

Factor Analysis (FA) [26]

FA is a statistical method used to describe variability among observed variables in terms of a potentially lower number of unobserved variables called factors. In other words, we wish to find a representation in a lower dimension by finding a way to represent the observed variables in a lower number of unobserved ones.

To perform FA we compute the maximum likelihood estimate of the factor loadings matrix Λ in the factor analysis model

$$M = \mu + \Lambda f + e \qquad (3)$$

where M is the m-by-n data matrix, μ is a constant vector of means, Λ is a constant m-by-n matrix of factor loadings, f is a vector of independent and standardized common factors, and e is a vector of independent specific factors. M, μ, and e are of length m. f is of length n.

Locality Preserving Projections (LPP) [25]

LPP generates linear projective maps by solving a variational problem that optimally preserves the neighborhood structure of a dataset. These projections are obtained by embedding the optimal linear approximations to the eigenfunctions of the Laplace Beltrami operator on the manifold. The resulting mapping corresponds to the following eigenvalue problem:

$$MLM^T w = \lambda MDX^T w \qquad (4)$$

where L is the Laplacian matrix, (i.e. $D - S$, where S corresponds to the similarity values defined, and D is a column matrix which reflects how important certain projections are). LPP is defined everywhere in the ambient space, not only on the training data points like Isomaps and LLE.

Non-linear dimensionality reduction methods

Isomap [28]

Isomap preserves pair-wise geodesic distances between data points. The geodesic distances between the data points x_i of M are calculated by constructing a neighborhood graph G, in which every data point x_i is connected with its k nearest neighbor's x_{ij} in the dataset M. The low-dimensional representations y_i of the data point's x_i in the low-dimensional space Y are computed by applying multidimensional scaling (MDS) [29] on the resulting m-by-m geodesic adjacency matrix.

Kernel PCA [27]

Kernel PCA computes the principal eigenvectors of the kernel matrix, rather than those of the covariance matrix. The reformulation of PCA in kernel space is straightforward, since a kernel matrix is similar to the inner-product of the data points in the high dimensional space that is constructed using the kernel function

Kernel PCA computes the kernel matrix K of the data-points x_i. The entries in the kernel matrix are defined by

$$k_{i,j} = (x_i, x_j) \qquad (5)$$

where $k_{j,i}$ is a kernel function. Then the kernel matrix K is centered by subtracting the mean of the features in traditional PCA.

To obtain the low-dimensional data representation, the data is projected onto the eigenvectors of the covariance matrix α_i. The entries of Y, the low dimensional representation are given by:

$$y_i = \left\{ \sum_{j=1}^{n} \alpha_1^j k(x_j, x_i), ..., \sum_{j=1}^{n} \alpha_d^j k(x_j, x_i) \right\} \qquad (6)$$

where α_l^j indicates the *jth* value in the vector α_l and k is the kernel function used in the computation of the kernel matrix. Since Kernel PCA is a kernel-based method, the mapping performed by Kernel PCA relies on the choice of the kernel function k. For this work we utilized a Gaussian kernel.

Laplacian Eigenmaps (Laplacian) [31]

Laplacian is another method that preserves the local properties of the manifold to produce a low-dimensional data representation. These local properties are based on the pair-wise distances between k nearest neighbors. Laplacian computes a low-dimensional representation of the data minimizing the distances between a data point and its k nearest neighbors.

The Laplacian algorithm first constructs a neighborhood graph G in which every data point x_i is connected to its k nearest neighbors. The weights of the edges on the connected data points of G are computed using the Gaussian kernel function, resulting in an adjacency matrix W. The computation of the degree matrix M and the graph Laplacian L of the graph W allows for the formulation the minimization problem as an eigenproblem. The low-dimensional representation Y can be found by solving the generalized eigenvalue problem:

$$Lv = \lambda Mv \tag{7}$$

for the d smallest nonzero eigenvalues, where d is the number of desired dimensions for Y. The d eigenvectors v_i corresponding to the smallest nonzero eigenvalues form the low-dimensional representation Y.

Locally-Linear Embedding (LLE) [30]

Similarly to Isomap, LLE constructs a graph representation of the data points and attempts to preserve the local properties of the data allowing for successful embedding of non-convex manifolds. In LLE, the local properties of the data are constructed by writing the local properties of the manifold around a data point x_i by writing the data point as a linear combination W_i of its k nearest neighbors x_{ij}, by fitting an hyperplane through the data point x_i and its nearest neighbors. Therefore finding the d dimensional data representation Y depends on minimizing the cost function

$$\phi(Y) = \sum_i (y_i - \sum_{j=1}^{k} w_{ij} y_{ij})^2 \tag{8}$$

where y_i is a data point in the low dimensional representation and w_{ij} is a reconstruction weight.

Classification algorithms

In order to help us determine the number of dimensions that we can reduce from our benchmark datasets, we decided to use classifiers for our comparative analysis of the performance of these dimensionality reduction methods on our benchmark datasets.

We selected Naïve Bayes (NB) as our linear classifier, Support Vector Machines (SVM) with a non-linear kernel function as our non-linear classifier, and C4.5 as a decision tree classifier. We opted to use these classification methods based on our literature review and previous research work [15, 17, 19 and 21]. As a brief summary, SVM has shown results that constantly depend on the nature of the dataset it is being applied to. C4.5 is also widely used in different applications and on research works, providing domain-specific results. NB due to its fast training and low computational cost is very popular and surprisingly accurate in many domains. A more detailed explanation behind our selection of these classification algorithms is presented in [17 and 21].

Approach and Experiments

All classification comparisons were performed in Weka 3.6.1 [32]. We utilized the default settings for all classifiers since we are performing a comparative analysis. We selected 67% of our data as the training set and an 'unseen' 33% test set for evaluation.

Six out of the eight (FA, KernelPCA, Isomap, LLE, LE, LPP) dimensionality reduction methods were investigated using the Matlab Tool box for dimensionality reduction [33]. For the remaining two (PCA and SVD), we utilized the standard Matlab functions.

For 'optimal' dimensionality estimation we decided to utilize the number of dimensions returned by standard PCA as presented in [41] and SVD's setting up a variance threshold between 96 and 99%. Tab. 1 presents these numbers of dimensions for all three datasets utilized.

Dataset	PCA Variance				SVD Variance			
	96%	97%	98%	99%	96%	97%	98%	99%
Solar [18]	42	46	51	58	58	74	99	143
INDECS [44]	94	106	121	143	215	239	270	319
ImageCLEF med [43]	79	89	103	126	193	218	253	307
Experiment Label	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***

Table 1. Number of dimensions selected for each dataset

For the non-linear methods that utilize neighborhood graphs we used between 6 and 12 as the number of nearest neighbors, and presented the best results since they varied by less than 0.001% of classification accuracy we decided to omit them from our presentation in this paper.

Mapping functions

Since we are planning on applying the best dimensionality reduction method on new data we will be receiving for the next five to ten years, we decided to simulate this scenario with our benchmark dataset.

In our experiments we performed the dimensionality reduction methods on 67% of our data and then map 'new' data points (the remaining 33% of our data) into the resulting low-dimensional representation that each dimensionality reduction method produces. We then feed these dimensionality reduced data points respectively as training and test sets to our classification algorithms.

For linear dimensionality reduction methods, the mapping of new data points is very straight forward since, for example for PCA and SVD you only have to multiply the new data points with the linear mapping matrix V.

As for non-linear dimensionality reduction methods, the mapping of new data points is not as straight forward. For Kernel PCA it is somewhat similar to the original PCA, but requires some additional kernel function computations as presented in [34]. For Isomaps, LLE, and LE we used kernel methods that have been presented in [35] and alternative approaches as shown in [36-40]

Experiments, observations and some conclusions

Fig. 2 shows classification accuracy of our three selected classifiers on the original non-reduced data sets and the 64 dimensionally reduced experiments (from Tab. 1, it can be seen that we investigated 8 sets of dimensions for each of the 8 dimensionality reduction methods). Figure 2 presents the original data (first row), then the 4 linear dimensionality reduction methods followed by the 4 non-linear.

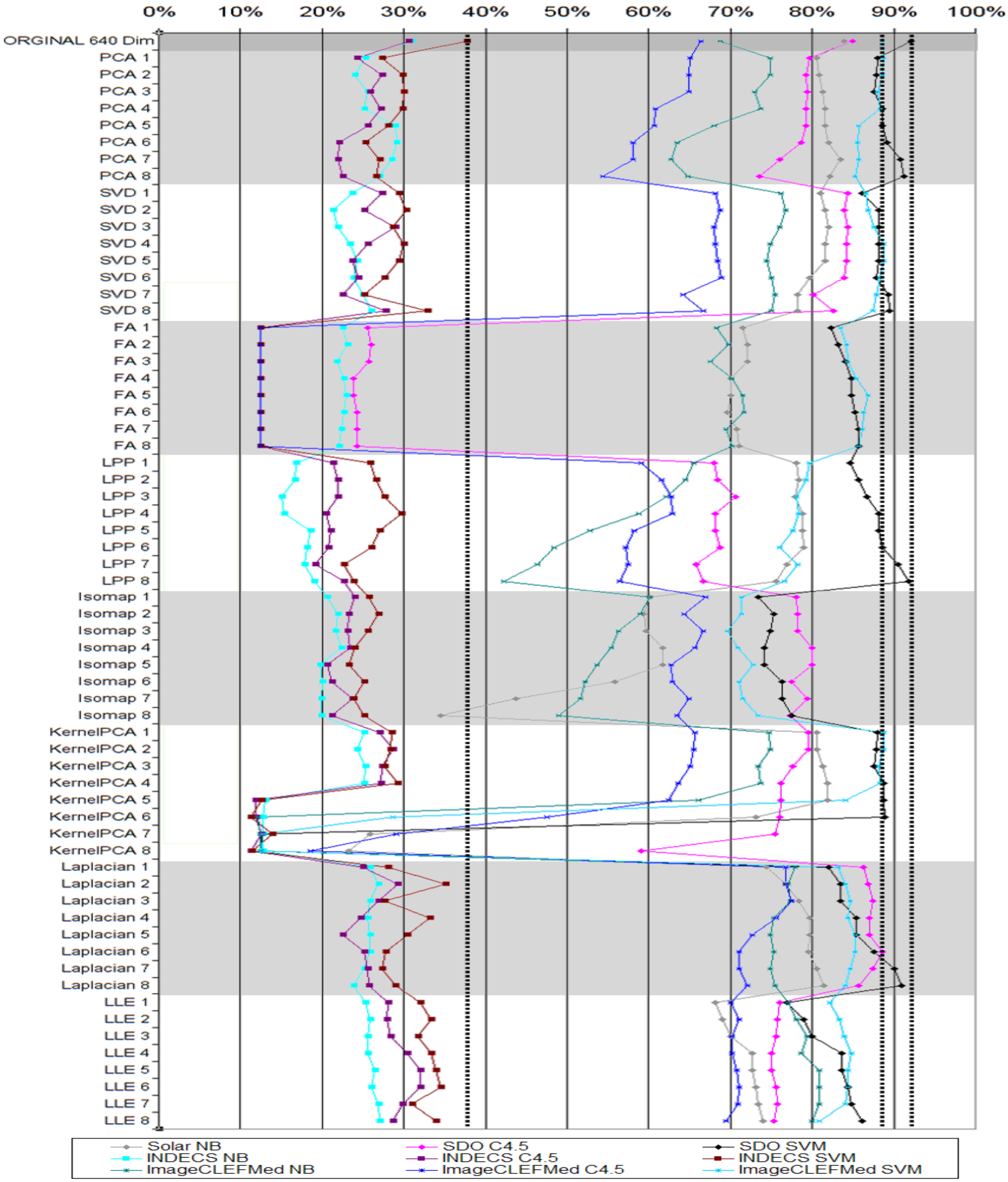


Figure 2. Classification accuracy of all dimensionality reduction methods for all 8 experiments per method for all datasets tested

Conclusions and Future Work

The first observation we can make is that our image parameters produced very bad results for a dataset (INDECS) that contains images very different from our own dataset (Solar [18]) and the other that contains images similar to ours (ImageCLEFMed). This clearly shows that using the right image parameters for a specific type of images is very important. We can also observe that SVM's produced most of the higher classification percentages for our Solar dataset and the ImageCLEFMed. To better show these occurrences we included bold dotted lines across fig. 2 for the highest classification results (SVM's) of the original datasets.

An interesting observation is that some combinations of classifiers and dimensionality reduction methods (e.g. LE and C4.5) actually produced better results than our original non-reduced dataset (for C4.5). We can also see a few of these combinations that produced very bad results (C4.5 and Factor Analysis) and others that dramatically drop in accuracy (KernelPCA for all classifiers),

The tree-based classifier performs very poorly with the Factor Analysis (FA) generated space by making very bad splitting decisions. Since C4.5 is a greedy algorithm (it never re-evaluates choice of attributes) it results in the accuracy dropping to 12%, which is equal to random labeling assignment. We can conclude that if you are using FA for your dimensionality reduction, it might be a good idea to stay away from decision tree based classifiers.

After looking at the drops in figure 2, KernelPCA showed very dramatic decrease in accuracy for a higher number of components. We attribute this drop to the partitioned space by kernel PCA, in a low number of dimensions this method achieves very good representation of our original data space, however when more dimensions are used, the method just produces more noise for the classifiers and damages their performance considerably.

Laplacian Eigenmaps (Laplacian) offer the most consistent results we have seen when applied to our Solar dataset. For all classifiers this dimensionality reduction method provides over 80% accuracy and for SVM it stays over 90% for the most part of fig. 2. The difference between the accuracy of each method is on average less than 10%, this supports the claim of how consistent this data representation is.

As we have mentioned in this paper, we are focusing on achieving dimensionality reduction in order to reduce storage costs, and if we have to sacrifice less than 3% in accuracy (especially at the 90% level) for more than 30% in storage, we are willing to take the hit in accuracy.

As we can see from the first half of fig. 2, the linear methods performed very consistently when the number of components increases. We can say that SVM is the most consistent and best performing classification method we utilized. Most of the non-linear dimensionality methods allow the classifiers to perform more consistently between them (bottom half of fig. 2). Even when doing badly, the classifiers accuracy stays on average within 10% of each other. Making these new dimensional spaces better suited for classification (on average) than the linear ones. We also show that our way of selecting the number of components, provides results on both sides of the fence for non-linear methods.

Out of all of the non-linear dimensionality reduction methods presented on figure 2, Laplacian and LLE are the only ones to show consistent classification accuracy improvement when compared to linear methods (with the exception of LPP). We theorize that since Laplacian preserves the properties of small neighborhoods around the data points; our benchmark dataset is benefited since many of our data points are highly correlated to each other, allowing Laplacian to preserve their local properties better. We find this very interesting considering that other works have shown that local dimensionality reduction techniques (i.e Laplacian, LLE), suffer from the intrinsic high dimensionality of the data [40, 44] and these type of methods perform poorly on certain datasets.

In general, as we have seen from our experiments and other works [16] that the applicability of dimensionality reduction methods is very dependent on the dataset used. For our purposes we think that the number of dimensions for our Solar dataset are safely determined by PCA and SVD with a variance between 96% and 99%, we see that we manage to approach both sides of the peak in classification accuracy in most cases, indicating that PCA approached the peak (highest accuracy) from the left side (low to high) and SVD's behaves the opposite way. The complexity of estimating dimensions this way in a much larger dataset might render these two techniques highly expensive since they rely on the calculation of Eigen vectors, but they seem to be accurate enough versus running experiments for all possible number of dimensions. The best performing methods for the solar data (in terms of higher percentage of accuracy) are PCA, LPP with 143 dimensions and Laplacian with 74 dimensions (table 2).

NB		C45		SVM	
ORGINAL	83.86%	Laplacian 6	88.56%	ORIGINAL	92.12%
PCA 7	83.49%	Laplacian 3	87.43%	LPP 8	91.74%
PCA 8	82.18%	Laplacian 7	87.43%	PCA 8	91.18%

Table 2. Top 3 results for each classifier and the solar dataset

Selecting anywhere between 42 and 74 dimensions provided very stable results for our dataset with all the methods presented in this paper, see figure 2. We conclude that for our current benchmark dataset we will be able to reduce our dimensionality around 90% from the originally proposed 640 dimensions. Considering that for the SDO mission we will have to store around 5.27 Gigabytes of data per day and 10,240 dimensions, a 90% reduction would imply savings of up to 4.74 Gigabytes per day.

Now that we have determined how many dimensions we can save by utilizing dimensionality reduction methods and which method to use, we can proceed along our path of building a CBIR system for the SDO mission.We can now focus on finding an indexing technique that can better suit retrieval of our data. Many works utilize some form of dimensionality reduction techniques in order to generate indexes and this will be our intended next step.

References

[1] Solar Dynamics Observatory [Online], Available: http://sdo.gsfc.nasa.gov/. [Accessed: Oct 20, 2011]

[2] 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, pp. 77-107.

[3] R. Datta, J. Li and Z. Wang “Content-based image retrieval – approaches and trends of the new age”. In ACM Multimedia. Singapore 2005.

[4] P. Moravec, and V. Snasel, “Dimension reduction methods for image retrieval”. In. ISDA ‘06. IEEE Computer Society, Washington, DC, pp. 1055-1060.

[5] J. Ye, R. Janardan, and Q. Li, “GPCA: an efficient dimension reduction scheme for image compression and retrieval”. In KDD '04. ACM, New York, NY, pp. 354-363.

[6] E. Bingham, and H. Mannila, “Random projection in dimensionality reduction: applications to image and text data”. In KDD '01. ACM, New York, NY, pp. 245-250.

[7] A. Antoniadis, S. Lambert-Lacroix, F. Leblanc, F. “Effective dimension reduction methods for tumor classification using gene expression data”. Bioinformatics, V. 19, 2003, pp. 563–570

[8] J. Harsanyi and C.-I Chang, “Hyper-spectral image classification and dimensionality reduction: An orthogonal subspace projection approach,” IEEE Trans. Geosci. Remote Sensing, vol. 32, July 1994, pp. 779–785.

[9] V. Zharkova, S. Ipson, et al. “Feature recognition in solar images”. Artificial Intelligence Review, 23(3), 2005 pp. 209–266.

[10] S. Christe, I.G. Hannah, et al. “Flare-Finding and Frequency Distributions”. ApJ, 677 (Apr 2008), pp. 1385–1394.

[11] P.N. Bernasconi, D.M. Rust, and D. Hakim. “Advanced automated solar filament detection and characterization code: description, performance, and results.” Sol. Phys 228 (May 2005) pp. 97–117.

[12] Savcheva A., Cirtain J., et.al. A Study of Polar Jet Parameters Based on Hinode XRT Observations. Publ. Astron. Soc. 59 pp771

[13] I. De Moortel and R.T.J McAteer. “Waves and wavelets: an automated detection technique for solar oscillations”, Sol. Phys. 223 (Sept. 2007), pp. 1–2.

[14] R. T. J. McAteer, P. T. Gallagher, et.al. “Ultraviolet oscillations in the chromosphere of the quiet sun”. ApJ 602, pp. 436–445.

[15] R. Lamb, “An information retrieval system for images from the TRACE satellite,” M.S. thesis, Dept. Comp. Sci., Montana State Univ., Bozeman, MT

[16] L.J.P. van der Maaten, E.O. Postma, and H.J. van den Herik. “Dimensionality reduction: a comparative review”. Tilburg University Technical Report, TiCC-TR 2009-005, 2009.

[17] J. M. Banda and R. Anrgyk “An Experimental Evaluation of Popular Image Parameters for Monochromatic Solar Image Categorization”. FLAIRS-23 (May 2010) pp. 380-385.

[18] SDO Dataset (MSU) [Online], Available: http://www.cs.montana.edu/angryk/SDO/data/ [Accessed: Oct 10, 2011]

[19] J.M. Banda and R. Angryk. “On the effectiveness of fuzzy clustering as a data discretization technique for large-scale classification of solar images.” In FUZZ-IEEE 2009, pp. 2019-2024

[20] B. B. Chaudhuri, S.Nirupam. “Texture segmentation using fractal dimension.” In TPAMI, vol. 17, no. 1, pp 72-77, 1995

[21] J.M. Banda, R.A. Angryk, P. Martens, "On the Surprisingly Accurate Transfer of Image Parameters between Medical and Solar Images", Proceedings of the 18th IEEE International Conference on Image Processing (ICIP-IEEE ’11), Brussels, Belgium, September 2011, pp. 3730-3733.

[22] A. Savcheva, J. Cirtain, et al. “A Study of Polar Jet Parameters Based on Hinode XRT Observations”. Publ. Astron. Soc. 59, 771

[23] K. Pearson, "On lines and planes of closest fit to systems of points in space" . Philosophical Magazine 2 (6) 1901, pp 559–572.

[24] C. Eckart, G. Young, "The approximation of one matrix by another of lower rank", Psychometrika 1 (3): 1936, pp 211–218.

[25] X. He and P. Niyogi, “Locality Preserving Projections,” In NIPS 2003. V 16. pp 153-160.

[26] D. N.Lawley, and A. E. Maxwell. “Factor analysis as a statistical method”. 2nd Ed. New York: American Elsevier, 1971.

[27] B. Schölkopf, A. Smola, and K.-R. Muller. Kernel principal component analysis. In ICANN97, Springer Lecture Notes in Computer Science, pp 583, 1997.

[28] J.B. Tenenbaum, V. de Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. Science, 290(5500):2319–2323, 2000.

[29] I. Borg, and P. Groenen. "Modern multidimensional scaling: theory and applications" (2nd ed.), Springer-Verlag New York, 2005

[30] L.K. Saul, K.Q. Weinberger, et al. Spectral methods for dimensionality reduction. In Semi-supervised Learning, Cambridge, MA, USA, 2006. The MIT Press.

[31] M. Belkin and P. Niyogi. “Laplacian Eigenmaps and spectral techniques for embedding and clustering”. In NIPS 2003, V. 14, pp 585–591.

[32] M. Hall, E. Frank, et al. “The WEKA Data Mining Software: An Update” SIGKDD Explorations, Volume 11, Issue 1. 2009

[33] Matlab Toolbox for Dimensionality Reduction [Online] Available: http://homepage.tudelft.nl/19j49/Matlab_Toolbox_for_Dimensionality_Reduction.html [Accessed: Oct 22, 2011]

[34] B. Schölkopf, A.J. Smola, and K.-R. M¨uller. “Nonlinear component analysis as a kernel eigenvalue problem”. Neural Computation, 10(5):1299–1319, 1998.

[35] Y. Bengio, J.-F. Paiement, et al. “Out-of-sample extensions for LLE, Isomap, MDS, eigenmaps, and spectral clustering”. In NIPS 2004 V 16.

[36] N.M. Rajpoot, M. Arif, and A.H. Bhalerao. Unsupervised learning of shape manifolds. In BMVC 2007.

[37] H. Choi and S. Choi. “Robust kernel Isomap”. Pattern Recognition, 40(3):853–862, 2007.

[38] V. de Silva and J.B. Tenenbaum. “Global versus local methods in nonlinear dimensionality reduc. In NIPS 2003, V15, pp 721–728

[39] A. Lima, H. Zen, Y. Nankaku, C. Miyajima, K. Tokuda, and T. Kitamura. “On the use of Kernel PCA for feature extraction in speech recognition”. IEICE Transactions on Information Systems, E87-D(12):2802–2811, 2004.

[40] K.Q. Weinberger, F. Sha, et al. “Graph Laplacian regularization for large-scale semi-definite programming”. In NIPS 2007, V19.

[41] A. Schlam, R. Resmini, D. Messinger, W. Basener. “A Comparison Study of Dimension Estimation Algorithms”. Technical Report, MITRE, July 19th, 2010

[42] Y. Bengio and Y. LeCun. “Scaling learning algorithms towards AI”. In Large-Scale Kernel Machines 2007, pp 321–360.

[43] W. Hersh, H. Müller, et al., “The consolidated ImageCLEFmed Medical Image Retrieval Task Test Collection”, Journal of Digital Imaging, volume 22(6), 2009, pp 648-655.

[44] A. Pronobis, B. Caputo, et al. “A discriminative approach to robust visual place recognition”. In IROS ‘06, pp. 3829-383