Dimension reduction for hyperspectral imaging using laplacian eigenmaps and randomized principal component analysis

Yiran Li
yl534@math.umd.edu

Advisor: Wojtek Czaja
wojtek@math.umd.edu

10/17/2014

Abstract

Hyperspectral imaging has attracted researchers' interests in recent years. Because of its high dimensionality and complexity, the reduction of dimension of hyperspectral data sets has become crucial in processing and categorizing the data. In this project, I will apply two methods of dimension reduction: laplacian eigenmaps and randomized principal component analysis in order to reduce the dimension of hyperspectral imaging data. Laplacian eigenmaps is a method developed in 2002 for dimension reduction. It is widely used and is essential in understanding dimension reduction. Randomized principal component analysis is a method that is developed later in 2008. It takes an approach different from laplacian eigenmaps, but is more efficient in dealing with large matrices. My goal is to implement these two methods to some hyperspectral data sets, and study in order to understand the differences and similarities between these two methods. This project will also assists me in understanding hyperspectral imaging and in analyzing hyperspectral data sets.

Introduction

With the development of hyperspectral sensors, hyperspectral imaging has drawn attention to researchers. In order to understand hyper spectral imaging, we need to learn about the notion of spectrum. Lights are generally described in terms of their wavelength. Different materials receive, transmit and reflect lights of different wavelengths. A spectrum of a certain material gives the percentage of light that is reflected by this material measured across a wide range of wavelengths. (Shippert, 2003). A hyperspectral imaging data set contains the spectrum of each pixel on the image. The data sets are presented as the following:

$$A(x, y, :) = \phi_{xy}$$

where x, y, are the x-coordinates and y-coordinates of the pixel in the image, and ϕ_{xy} is an n-dimensional vector that gives the spectrum of the pixel at (x, y), assuming that there are n bands of wavelengths at which we measure the reflectance. If we take images of 1000 pixels, and use $n = 100$ bands of wavelengths, we would end up with a data set of size 10^8. Hence it is easy to see that hyperspectral data set can be large and thus difficult to analyze. Because of the large magnitude, there have been great incentives to reduce the dimensionality of hyperspectral data, in order for us to simplify the data complexity to enable calculations. Dimension reduction may also help reveal some underlying structures in data for the purposes of target or anomaly detection, and give us better classification and segmentation of the data set.

Various algorithms have been developed in recent years for the purpose of dimension reduction. A list of them are: Principal Component Analysis (PCA), Probabilistic PCA, Factor Analysis (FA), Classical multidimensional scaling (MDS), Linear Discriminant Analysis (LDA), Isomap, Local Linear Embedding (LLE), Local Tangent Space Alignment (LTSA), Fast Maximum Variance Unfolding (FastMVU), Kernel PCA. (Delft University) These algorithms may serve for different purposes, and thus they may differ in operation costs yet be of equal interests to different groups of people.

Approach

I will implement two algorithms of dimension reduction: laplacian eigenmaps and randomized principal component analysis. Laplacian eigenmaps is a widely used and well-developed method for dimension reduction. Implementing laplacian eigenmaps is an essential step to take in learning dimension reduction algorithms. The second method that I will use is randomized PCA. It is developed after laplacian eigenmaps and the method is more efficient under certain circumstances. The two algorithms are presented below. I will introduce them in combination with my application data sets, hyperspectral imaging data sets.

Laplacian eigenmaps

In the paper Laplacian Eigenmaps for Dimensionality Reduction and Data Representation by Mikhail Belkin and Partha Niyogi, they introduced the method of laplacian eigenmaps as the following.

Let n be the number of data points in, in our case the number of pixels in the image, and let x_i denote the ith data point.

Step 1: Constructing the Adjacency Graph

We try to construct a weighted graph with n nodes, and a set of edges connecting neighboring points. There are two ways of constructing edges. The first way is to connect the nodes that are within ϵ of each other, for some real number $\epsilon > 0$. This means: x_i, x_j are connected if

$$|x_i - x_j|^2 < \epsilon.$$

. The second way is to connect the m nearest nodes of any node.

Step 2: Choosing the weights

There are two ways of choosing the weight on the edge. Let W_{ij} denote the weight function between points x_i, x_j. The first way is to define the weight on edge connecting point x_i, x_j to be the heat kernel. Indeed, let

$$W_{ij} = e^{-|x_i - x_j|^2/t}$$

where t is a parameter to be specified by the user. The second way is the simple minded way of choosing the weights: let

$$W_{ij} = 1$$

if two nodes are connected and

$$W_{ij} = 0$$

otherwise.

Step 3: Compute eigenvalues and eigenvectors

We will compute the eigenvalues and eigenvectors for the generalized eigenvector problem:

$$Lf = \lambda Df$$

where W is the weight matrix defined in step 2, D is diagonal weight matrix, with $D_{ii} = \sum_{j=1}^{n} W_{ji}$, and $L = D - W$. Let f_0, f_1 ... f_{n-1} be the solutions of equation $Lf = \lambda Df$, ordered such that

$$0 <= \lambda_0 <= \lambda_1 <= ... <= \lambda_(n-1)$$

Then the first m eigenvectors (excluding f_0),

$$f_1, f_2, ..., f_m$$

are the desired vectors for embedding our high dimensional data (n- dimensional) in m-dimensional Euclidean space.

Randomized Principal Component Analysis

In the paper A Randomized Algorithm for Principal Component Analysis by Vladimir Rokhlin, Arthur Szlam, and MarkTygert, the method randomized PCA was introduced. In our case, we assume that there are m pixels in the image. Ordering the pixels as x_1, x_2, ..., x_m, we form a matrix A, such that the ith row of A,

$$A_i = \phi_i,$$

where ϕ_i is the spectrum (n-dimensional vector) of pixel x_i. Assume that we want to reduce the dimension of A from m to k.
The algorithm in Rokhlin's paper is presented as the following: A general way of constructing the best rank-k approximation to a real mxn matrix A uses single value decomposition:

$$A = U\Sigma V^T,$$

where U real unitary mxm matrix, V is real unitary nxn matrix, and Σ is real mxn diagonal matrix with nonnegative, non increasing diagonal entries. The best rank-k approximation of A is given by

$$A \approx \tilde{U}\tilde{\Sigma}\tilde{V}^T,$$

where $\tilde{U}$ is the leftmost mxl block of U, $\tilde{\Sigma}$is the kxk upper left block of Σ, $\tilde{V}$ leftmost nxk block of V. This approximation is considered the best because it minimizes the spectral norm $||A - B||$ for a rank-k matrix $B = \tilde{U}\tilde{\Sigma}\tilde{V}^T$,. In fact,

$$||A - \tilde{U}\tilde{\Sigma}\tilde{V}^T|| = \sigma_{(k+1)},$$

where $\sigma_{(k+1)}$ is the $(k+1)^t h$ greatest singular value of A. The randomized PCA generates B such that

$$||A - B|| <= Cm^{(1/(4i+2))}\sigma_{(k+1)}$$

with high probability $(1 - 10^(-15))$, where i is specified by user, and C depends on parameters of algorithm. Let us choose $l > k$ such that $l <= m - k$.

Step 1

Generate a real lxm matrix G whose entries are independently, identically distributed normal Gaussian random variables, and compute

$$R = G(AA^T)^i A.$$

?

Step 2

Using SVD, form a real nxk matrix Q whose columns are orthonormal, such that

$$||QS - R^T|| <= \rho_{(k+1)}$$

for some kxl matrix S, where $\rho_{(k+1)}$ is the $(k+1)^th$ greatest singular value of R.

Step 3

Compute

$$T = AQ.$$

Step 4

Form an SVD of T:

$$T = U\Sigma W^T,$$

where U is a real mxk matrix whose columns are orthonormal, W is a real kxk matrix whose columns are orthonormal, Σ is a real diagonal kxk matrix with nonnegative diagonal entries.

Step 5

Compute

$$V = QW.$$

In this way, we get U, Σ, V as desired.
In implementing laplacian eigenmaps, we could end up with disconnected graphs because of the way we construct edges. If that is the case, I will run the algorithm for each disconnected part and combine the results for each disconnected graph to get final results. It is also possible that we get different results when using different weight functions and connecting the graph in different ways, so I plan to experiment with two weights functions as described above, and choose a reasonable range of ϵ s to connect the graph based on the distances between data points in my particular data sets.
In implementing randomized PCA, I still need to look into more details on how to find Q using single value decomposition at step 2. But because there are reference codes that I could refer to for SVD's, I will learn from the accomplished codes in tackling my own problems. I could also use while loops in my code to search for Q that satisfy the bound. There might be memory issues when my data sets get large. If that happens, I will consult my advisor and get access to the super computer in Norbert Weiner Center to use.

Implementation

I will implement these algorithms in Matlab, on my personal laptop and on the computers in the computer lab in math department. The hyperspectral data will be around hundreds of pixels and 10s to hundreds of bands, and for the ones I?ve downloaded so far, they are in reasonable size. I will try to increase data size after testing on small cases, and if there is trouble computing due to large data size, I will ask for access to more powerful computers.

Data base

Here is a list of data bases that I will be using: 12 Band Moderate Dimension Image: June 1966 aircraft scanner Flightline C1 (Portion of Southern Tippecanoe County, Indiana); 220 Band Hyperspectral Image: June 12, 1992 AVIRIS image Indian Pine Test Site 3 (2 x 2 mile portion of Northwest Tippecanoe County, Indiana); 220 Band Hyperspectral Image: June 12, 1992 AVIRIS image North-South flight line (25 x 6 mile portion of Northwest Tippecanoe County, Indiana). These data sets are the hyperspectral imaging taken across the United States, and the result of dimension reduction of these data sets are convenient to validate. These data sets are publicaly available online, and they are of reasonable size to test my algorithms.

Validation

There is a matlab toolbox developed by people from Delft University for dimension reduction. It contains numerous methods for dimension reduction, and is publically available. Hence I can use the toolbox to reduce dimension of the same data sets as the ones I tested my algorithms on, and compare the results I get using the toolbox with the result I get implementing my algorithms. Comparing the results I get using two different algorithms on the same data sets might also help me validate my methods. For randomized PCA, since the author provided an error bound to the result, I can check the error bound once I implemented the algorithm.

Testing

Implementing two different algorithms on the new databases would lead to interesting results to be compared. Plotting the reduced dimensional space can help us visualize and understand the data with reduced dimension. We expect to get similar results for two different methods, but it is possible that running time differs, and that one algorithm works better than the other under certain circumstances.

Project Schedules and milestones

The timeline for my project and the corresponding milestones in each period are the following.

- October 17th: finish and revise project proposal.
- October to November, 2014: Implement and test laplacian eigenmaps on databases, prepare for implementation of randomized PCA. Get results for the reduced dimensions of images in databases using laplacian eigenmaps.
- December, 2014: finish midyear report and presentation.
- January to March: Implement and test randomized PCA, compare two methods in various situations. Get results for the reduced dimensions of databases using randomized PCA.
- April to May: finish final presentation and final report deliverables.

I will be able to deliver the following items of my project by the end of next spring semester.

- Presentation of data sets with reduced dimensions of both algorithms;
- Comparison charts in terms of running time and accuracy of two different methods;
- Comparison charts with other methods that are available from the DR matlab toolbox; Databases;

- Matlab codes for both algorithms;
- Presentations;
- Proposal;
- Mid-year report;
- Final report.

References

[1] Shippert, Peg. *Introduction to Hyperspectral Image Analysis*. Online Journal of Space Communication, issue No. 3: Remote Sensing of Earth via Satellite. Winter 2003.

[2] Rokhlin, Vladimir; Szlam, Arthur; Tygert, Mark. *A Randomized Algorithm for Principal Component Analysis*. SIAM Journal on Matrix Analysis and Applications Volume 31 Issue 3. August 2009.

[3] Belkin, Mikhail; Niyogi, Partha. *Laplacian Eigenmaps for Dimensionality Reduction and Data Representation*. Neural Computation, vol 15. Dec. 8th, 2002. Web.

[4] Matlab Toolbox for Dimension Reduction. Delft University. Web. Oct. 6th, 2014.