AMATH 582 Homework 5

Tomas Perez

March 18, 2021

Abstract

This paper illustrates a use case of data-based algorithms in image analysis. In particular, Dynamic Mode Decomposition is used to separate two videos into foreground and background videos.

1 Introduction and Overview

Two videos are provided as data sets. A clip from the F1 Monaco Grand Prix in Monte Carlo, and a clip of a psychopath skiing of the face of a cliff. The videos are first converted into matrices that can be used in MATLAB. We then use Singular Value Decomposition and obtain the DMD's approximate low-rank reconstruction and the approximate sparse reconstruction.

2 Theoretical Background

Consider our m data vectors, of length n, that represent different snapshots of the dataset $X_{(mxn)}$ over time, t. We data can be separated into two matrices,

$$X_1 = [x_1, ..., x_{m-1}] \tag{1}$$

$$X_2 = [x_2, ..., x_m] \tag{2}$$

The goal of Dynamic Mode Decomposition (DMD) is to find the best matrix A such that $x_{i+1} = Ax_i$. Applying our two X_i matrices, we get the following:

$$X_2 = AX_1 \tag{3}$$

and the psuedo-inverse,

$$A = X_2(X_1)' \tag{4}$$

We then compute the SVD of X_1 truncated to rank r,

$$X_1 = U_r \Sigma_r V_r^* \tag{5}$$

With this we can get the rank r approximation of A,

$$A_r = U^* AU = U^* X_2 V \Sigma^{-1} \tag{6}$$

From here we can compute the eigenvector matrix K and eigenvalue matrix D of the r-rank approximation and get the DMD modes,

$$\Phi = X_2 V \Sigma^{-1} W \tag{7}$$

Now we can get the approximate DMD solution at all future times,

$$X_{DMD}(t) = \sum_{k=1}^{K} b_k(0)\phi_k(x)e^{wt} = \Phi e^{Dt} b \tag{8}$$

3 Algorithm Implementation and Development

Algorithm 1: DMD

Load and reshape the video data by frames
Create X_1 and X_2 matrices
Perform SVD and get the U,Σ, and V matrices
Truncate the decomposed matrices to rank r
Compute the low-rank approximation
Obtain the eigenvalues and eigenvectors
Compute DMD modes
Construct the low-rank X_{DMD} and sparse X_{DMD}. Add contrast to the sparse matrix if necessary.
Display the foreground video extracted by the low-rank DMD and the background extracted by the sparse DMD

4 Computational Results

Figure 1 and Figure 2 show the singular values and the percentage variance explained. For both the skiing and F1 videos, it is clear that the first singular value is by far the most dominant one. Therefore the rank used in this analysis is $r = 1$. Figures 3-8 show the DMD analysis of the 100th frame of each video. Each figure displays the image from the original video, the background extracted from the low-rank DMD, and the foreground extracted from the sparse DMD. One issue that is evident in these videos is the lack of contrast. To adjust for this, the videos are "scaled" by adding positive values to the sparse DMD matrix. The outcome clearly improves with this addition. The separation of the skiing video is successful, but it is much harder to tell given the size of the skier in the video.

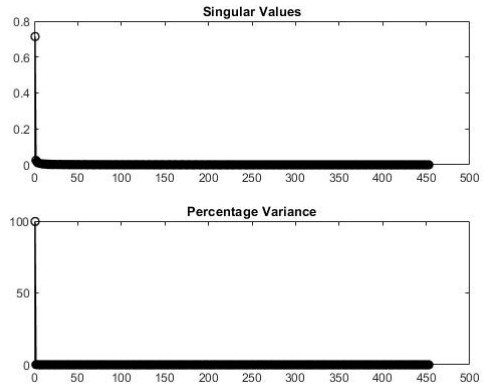

Figure 1: Skiing Video: Singular Values

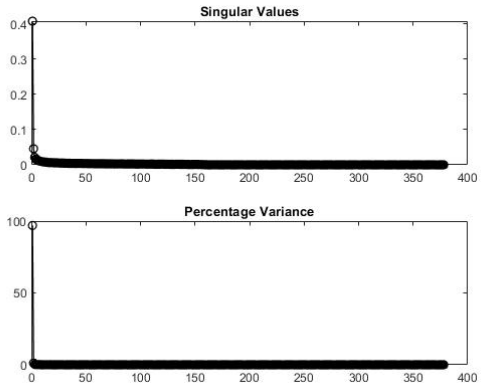

Figure 2: F1 Video: Singular Values

Figure 3: Skiing Video: DMD of Frame 100 (contrast=0)

Figure 4: F1 Video: DMD of Frame 100 (contrast=0)

Figure 5: Skiing Video: DMD of Frame 100 (contrast=50)

Figure 6: F1 Video: DMD of Frame 100 (contrast=50)

Figure 7: Skiing Video: DMD of Frame 100 (contrast=100)

Figure 8: F1 Video: DMD of Frame 100 (contrast=100)

5 Summary and Conclusions

This study illustrates an application of dynamic mode decomposition on video analysis. The videos can successfully be separated into foreground and background, though separating the foreground of the skiing

video proved to be more difficult than the F1 video. The addition of some positive values to the sparse matrix further improved the results.

Appendix A MATLAB Functions

- `V = VideoReader(filename)` creates object v to read video data from the file named filename.
- `B = reshape(A,sz)` reshapes A using the size vector, sz, to define size(B). For example, reshape(A,[2,3]) reshapes A into a 2-by-3 matrix. sz must contain at least 2 elements, and prod(sz) must be the same as numel(A).
- `[V,D]=eig(A)` returns diagonal matrix D of eigenvalues and matrix V whose columns are the corresponding right eigenvectors, so that A*V = V*D
- `D=diag(v)` returns a square diagonal matrix with the elements of vector v on the main diagonal
- `[U,S,V]=svd(A)` performs singular value decomposition of matrix A, such that A=U*S*V
- `Y = uint8(X)` converts the values in X to type uint8. Values outside the range $[0,2^8 - 1]$ map to the nearest endpoint.
- `imshow(I)` displays the grayscale image I in a figure. imshow uses the default display range for the image data type and optimizes figure, axes, and image object properties for image display.

Appendix B MATLAB Code

```
close all; clear; clc;
Video=VideoReader('ski_drop_low.mp4');
%Video=VideoReader('monte_carlo_low.mp4');
for i=1:1:Video.NumFrames
    grayFrame=rgb2gray(read(Video,i));
    Frame=reshape(grayFrame,Video.Width*Video.Height,1);
    Frames(:,i)=double(Frame);
end
t2=linspace(0,Video.CurrentTime, Video.NumFrames+1); t=t2(1:end-1);
dt=t(2)-t(1);
%% DMD
X1=Frames(:,1:end-1); X2=Frames(:,2:end);
[U,S,V]=svd(X1,'econ');
lambda=diag(S).^2;
figure(1) %singular values
subplot(2,1,1), plot(diag(S)/sum(diag(S)),'ko-','Linewidth',[1])
title('Singular Values')
subplot(2,1,2), plot(lambda/sum(lambda)*100,'ko-','Linewidth',[1])
title('Percentage Variance')

r=1; % rank 1
U=U(:,1:r); S=S(1:r,1:r); V=V(:,1:r);
A=U'*X2*V*diag(1./diag(S)); % low rank aproximation
[eigVec,eigVal]=eig(A);
Phi=X2*V/S*eigVec; % DMD modes
% low-rank DMD
mu=diag(eigVal);
omega=log(mu)/dt;
x1=X1(:,1); y0=Phi\x1;
modes=zeros(r,length(t));
modes(:,1)=y0;
for j=1:length(t)-1
    modes(:,j)=(y0.*exp(omega*t(j)));
end
Xdmd=Phi*modes;
% sparse DMD
Xsparse=(Frames-abs(Xdmd)); %add positive values for more contrast
R=Xsparse.*(Xsparse<0.01);
Xdmd=R+abs(Xdmd);
Xsparse=Xsparse-R;
%%
k=100;
figure(2)
subplot(3,1,1)
imshow(reshape(uint8(Frames(:,k)),Video.Height,Video.Width))
title('Original')
subplot(3,1,2)
imshow(reshape(uint8(Xdmd(:,k)),Video.Height,Video.Width))
title('Low-rank DMD')
subplot(3,1,3)
imshow(reshape(uint8(Xsparse(:,k)),Video.Height,Video.Width))
title('Sparse DMD')
```