Coding Project 5: Background Subtraction through Dynamic Mode Decomposition

Wietse Vaes

Abstract

In videos we always have a main focus, the foreground, and the rest, the background. We know the difference, but could a computer? Using a new decomposition, the dynamic mode decomposition, we can identify the background from foreground by it's importance. Using the singular value decomposition and a bit of transforming, one can find a new way of describing data using a different set of basis, given a couple of restrictions.

Introduction In short, dynamic mode decomposition (DMD) decomposes a data set into time dependent functions using eigenvectors and eigenvalues. These eigenvectors form a basis, which is a very useful fact. In this scientific letter we will focus on subtracting backgrounds, but a lot more can be done using DMD. Most are related to scaling structure as the eigenvalues will give a sort of importance of that time scale. Later on we will see that a transformation of the eigenvalues will be multiplied by your time variable t. Therefore we can find large-scale and small-scale structures in movies. We can also identify backgrounds and foregrounds. This can all be done without knowing underlying models and is thus rather effective in finding movements.

In the following short paper we will go through finding the decomposition by finding a way to approximate the evolution of the first time steps to the last time steps. Based on an assumption that a Koopman operator exists between time step, We can approximate a way of getting from one time step to another. Using the eigenvectors and eigenvalues of the process, we get the DMD and thus will be able to find the background of the video.

Theoretical Background As this is a short scientific letter, we only talk about the DMD procedure and separating foreground and background. Say we have M equally spaced captures of a discrete function in time with N discrete spatial points. We can represent this data set as

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_M \end{bmatrix},$$

where $\mathbf{x}_j$ are the values of the discrete spatial points at time j in a vector form. Define the following matrix as well

$$\mathbf{X}_j^k = \left[\mathbf{x}_j, \mathbf{x}_{j+1}, \ldots, \mathbf{x}_k\right].$$

Assume now that a linear, time-independent operator, namely the Koopman operator, A exists such that

$$\mathbf{x}_{j+1} = A\mathbf{x}_j.$$

We then have

$$\mathbf{X}_1^{M-1} = \begin{bmatrix}\mathbf{x}_1 & A\mathbf{x}_1 & \ldots & A^{M-2}\mathbf{x}_1\end{bmatrix}.$$

Say now that $\mathbf{U} \in \mathbb{C}^{N\times K}$,$\mathbf{\Sigma} \in \mathbb{C}^{K\times K}$ and $\mathbf{V} \in \mathbb{C}^{M-1\times K}$ are the K reduced SVD approximation to the rank of $\mathbf{X}_1^{M-1}$. It is easy to see that we could represent $\mathbf{x}_M$, the last point, by using the other snapshots as basis:

$$\mathbf{x}_M = \sum_{j=1}^{M-1} b_j\mathbf{x}_j + \mathbf{r},$$

where $\mathbf{r}$ is orthogonal to all $\mathbf{x}_j$ and b_j are unknown. We now define

$$\mathbf{X}_2^M = \begin{bmatrix}A\mathbf{x}_1 & A^2\mathbf{x}_1 & \ldots & A^{M-1}\mathbf{x}_1\end{bmatrix} = A\mathbf{X}_1^{M-1}.$$

Aknowledge that $\mathbf{X}_1^{M-1}$ are the first $M-1$ snapshots and $\mathbf{X}_2^M$ the last $M-1$, and they have $M-2$ of the same points. Thus,

$$\mathbf{X}_2^M = \mathbf{X}_1^{M-1}\mathbf{S} + \mathbf{r}e_{M-1}^*,$$

where the jth column of $\mathbf{S}$ is comprised of zeros except one 1 on the jth row. The last column is made up out of the unknown b_j. Here, e_{M-1} is the $M-1$th unit vector. Computing S seems impossible as we do not know the values of b_j, but we can magically approximate it by using the svd decomposition of A. We will approximate it by

$$\tilde{\mathbf{S}} = \mathbf{U}^*\mathbf{X}_2^M\mathbf{V}\mathbf{\Sigma}^{-1}.$$

Assuming this also has rank K, denote the $\mathbf{U}$ transformed eigenvectors of $\tilde{\mathbf{S}}$ as ψ_k and natural log of the eigenvalues divided by our sampling rate as ω_k, with $k = 1, \ldots, K$. We can then approximate our discrete spatial points in a time-continuous function as the Dynamic mode decomposition vectors

$$\mathbf{X}_{DMD}(t) = \sum_{k=1}^{K} b_k(0)\psi_k e^{\omega_k t},$$

where $b_k(0)$ can approximately be found found using the initial conditions and solving the linear system there.

The background can now be found by using only the lowest, in absolute value as these eigenvalues and vectors are complex, eigenvalues. This is a low-rank approximation of sorts. With this, we can then subtract this background, in absolute value, from our data to find the foreground.

Results After implementing this into MATLAB, we were able to subtract the moving dot (the foreground) from the background. This, however, was not perfect. In the background we see a constant image, which is what we want here. However, we can clearly see the ending of the trajectory of the ball. The beginning is however not very visible. Thus, this might be fixed by calculating the background using the original clip and the clip in reversed, and then reversing it back, and the averaging both resulting clips. The foreground has the same problem, the dot is clearly seen, but the ending of the trajectory is also very clearly seen. Once again, we can just calculate the foreground using the original clip and reversed clip, then reversing the foreground of this result and average it. We can also see sort of a trail on the ball while it's moving.

Conclusion Using data collected with a constant time step we were able to go from a time discrete set to a time continuous approximation, while not using something such as interpolation, but using dynamic mode decomposition. Based on the fact that these time points are connected with a linear, time-independent operator A, we were able to base a function from a new orthogonal set which might be able to predict the future movement as well without knowing a lot about underlying theoretical models. A lot of applications can be come from using these new basis, but we focused on subtracting the background from the foreground. This did a good job, but still left artifacts from the trajectory of the foreground. It is still, however, a magical thing to see.

Acknowledgment I'd like to acknowledge dr. Amin Rahman for the interesting projects.