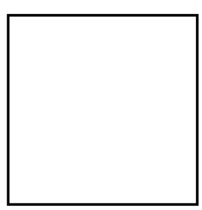
Applications of Eigenvalues and Eigenvectors in Computer Science

MTH3102: Intro to Linear Alebra, Section 01
Professor: Dr. Munevver Subasi
Eric Pereira

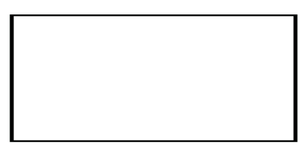
April 14, 2019

Eigenvectors are vectors whose directions are consistent when a linear transformation occurs. This linear transformation is defined by the eigenvalue, a scalar, such that the equation is of the form: $A\vec{v} = \lambda\vec{v}$ Where A is an $n \times n$ matrix, $\vec{v}$ is the eigenvector, a $1 \times n$ matrix, and λ is the eigenvalue, a scalar. There are quite a few really important and/or applications for this in computer science, such as in cases like computer vision, and search engine usage.

Computer vision offers many uses of eigenvalues and eigenvectors, mainly because it is important to stretch and/or compress (types of linear transformations) images in certain ways. A simple example of this can be seen in word processors, where you can place an image in the word processor and you may morph the image by stretching it vertically or horizontally. In this way the image is being scaled in a way where it was not initally designed to, in order to continue to make this image look relatively similar it is morphed and strecthed using eigenvalues and eigenvectors. Here is an example with a square image being scaled in such a way:



original image



scaled image

The example shows how the eigenvectors do not change direction when a linear transformation occurs, this linear tranformation being a scaling with a factor of 1.5 horizontally and 0.75 vertically. Of course this is one of the most basic ways to physically display a common use of eigenvectors and eigenvalues, but this simple idea is crucial in other parts of computer vision where proper scaling is important, in even more complex forms, such as facial recognition software. Facial recognition is used to unlock phones for example, and the phone camera has to scan a face and certain specific facial features. However, it also has to scale the face as when people raise the phone and look at it it is not always possible hold it at *exactly* the same distance each time. Using linear transformations through the use of eigenvectors and eigenvalues it is possible to hold your phone different distances away from a face and still have the phone recognize the right user and unlock or remain locked.