Video

Bookmark this page

How Does PCA Work?

actually see linked to in the dive

important details or important data.

allows you to intelligently reduce the dimensionality

of your dataset without having to worry too much

1:19 / 1:19 1.25x HD

Video

Download video file

Transcripts

Download SubRip (.srt) file

Download Text (.txt) file

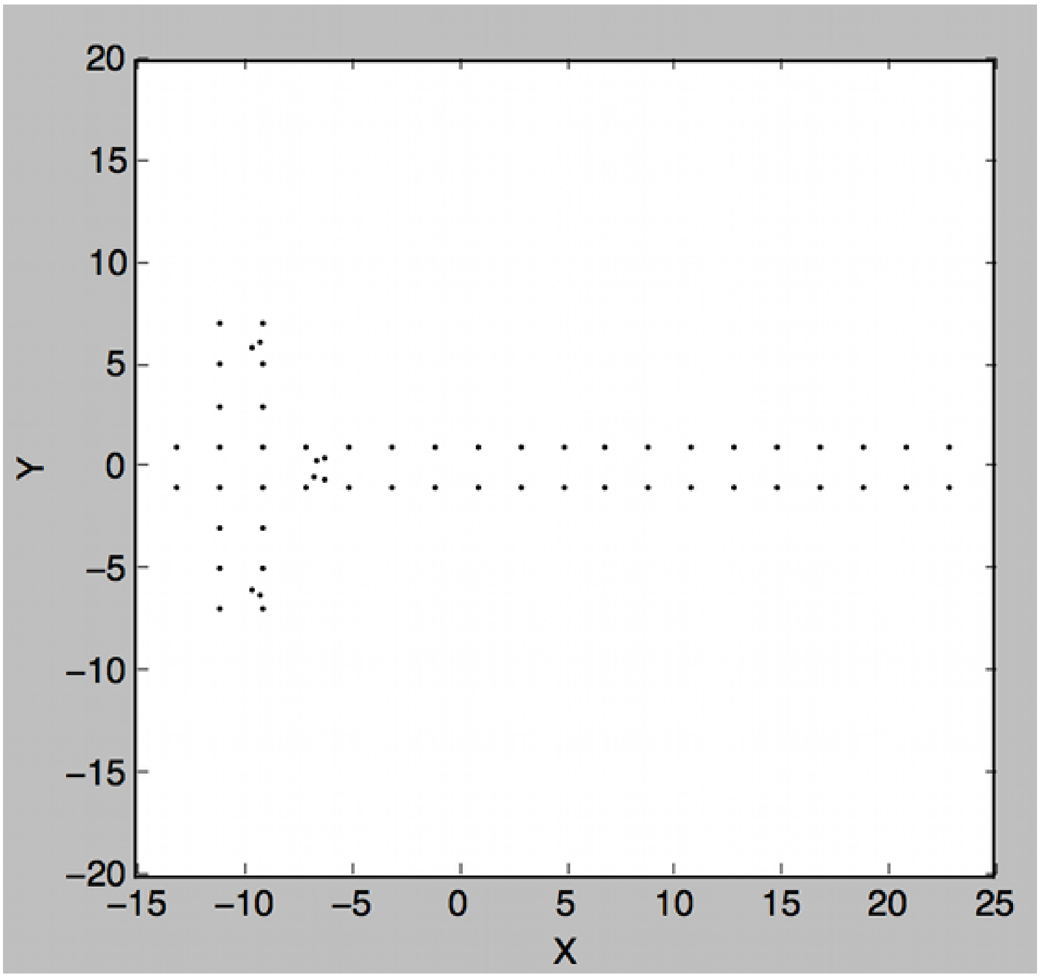
PCA is one of the most popular techniques for dimensionality reduction, and we recommend you always start with it when you have a complex dataset. It models a linear subspace of your data by capturing its greatest variability. Stated differently, it accesses your dataset's **covariance** structure directly using matrix calculations and eigenvectors to compute the *best* unique features that describe your samples.

An iterative approach to this would first find the center of your data, based off its numeric features. Next, it would search for the direction that has the most variance or widest spread of values. That direction is the principal component vector, so it is then added to a list. By searching for more directions of maximal variance that are orthogonal to all previously computed vectors, more principal component can then be added to the list. This set of vectors form a new feature space that you can represent your samples with.

On Dimensions, Features, and Views

Each sample in your dataset represents an observable phenomenon, such as an object in the real world. Each feature in your dataset tells you details about your samples. Recall from earlier chapters that features and views are synonymous terms; this isn't accidental! Just like looking at an object from different *views* gives you more information about the object, so too does examining a sample from different *features*. Similar or correlated features will produce an "overlapped" view of your samples, the same way similar views of an object also overlap.

PCA ensures that each newly computed view (feature) is *orthogonal* or linearly independent to all previously computed ones, minimizing these overlaps. PCA also orders the features by importance, assuming that the more variance expressed in a feature, the more important it is. In our telephone pole example, the frontal view had more variance than the bird's-eye view and so it was preferred by PCA.



With the newly computed features ordered by importance, *dropping* the least important features on the list intelligently reduces the number of dimensions needed to represent your dataset, with minimal loss of information. This has many practical uses, including boiling off high dimensionality observations to just a few key dimensions for visualization purposes, being used as a noise removal mechanism, and as a pre-processing step before sending your data through to other more processor-intensive algorithms. We'll look at more real life use cases in the next unit.