



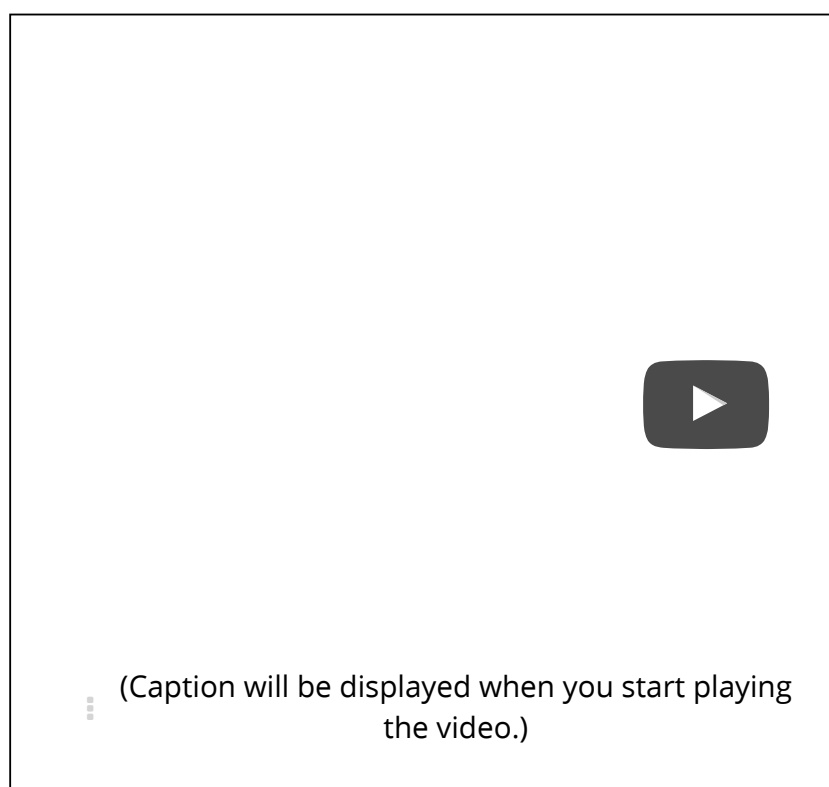
[Course](#) > [4. Transforming Data](#) > [Lecture: PCA](#) > Video

Video

🔖 [Bookmark this page](#)

When Should I Use PCA?

[Start of transcript. Skip to the end.](#)



PCA has many applications, since it orders your features by importance.

The last few features especially in larger data sets that come

from the real world, they'll be dominated by noise and that's the reason why they would even deprioritize.

PCA can therefore be used as a noise removal method.



Video

[Download video file](#)

Transcripts

[Download SubRip \(.srt\) file](#)

[Download Text \(.txt\) file](#)

PCA, and in fact *all* dimensionality reduction methods, have three main uses:

- To handle the clear goal of reducing the dimensionality and thus complexity of your dataset.

- To pre-process your data in preparation for other supervised learning tasks, such as regression and classification.
- To make visualizing your data easier, since we can only perceive three dimensions simultaneously.

According to Nielson Tetrad Demographics, the group of people who watch the most movies are people between the ages of 24 through 35. Let's say you had a list of 100 movies and surveyed 5000 people from within this demographic, asking them to rate all the movies they've seen on a scale of 1-10. By having considerably more data samples (5000 people) than features (100 ordinal movie ratings), you're more likely to avoid the curse of dimensionality.

Having collected all that data, even though you asked 100 questions, what do you think truly is being measured by the survey? Overall, it is the collective movie preference per person. You could attempt to solve for this manually in a supervised way, by break down movies into well-known genres:

- Action
- Adventure
- Comedy
- Crime & Gangster
- Drama
- Historical
- Horror
- Musicals
- Science Fiction
- War
- Western
- etc.

Being unsupervised, PCA doesn't have access to these genre labels. In fact, it doesn't have or care for *any* labels whatsoever. This is important because it's entirely possible there wasn't a single western movie in your list of 1000 films, so it would be inappropriate and strange for PCA to derive a 'Western' principal component feature. By using PCA, rather than you creating categories manually, it *discovers* the natural categories that exist in your data. It can find as many of them as you tell it to, so long as that number is less than the original number of features you provided, and as long as you have enough samples to support it. The groups it finds are the principal components, and they are the best possible, linearly independent combination of features that you can use to describe your data.

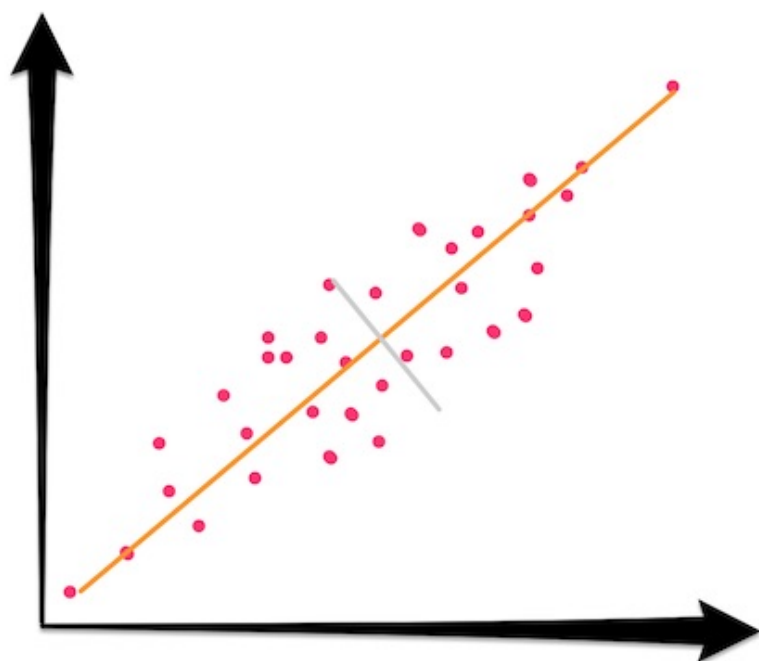
One warning is that again, being unsupervised, PCA can't tell you exactly what the newly created components or features *mean*. If you're interested in how to interpret your principal components, we've included two sources in the dive deeper section to help out with that and highly recommend you explore them.

Once you've reduced your dataset's dimensionality using PCA to best describe its variance and linear structure, you can then transform your movie questionnaire dataset from its original [1000, 100] feature-space into the much more comfortable, principal component space, such as [1000, 10]. You can visualize your samples in this new space using an Andrew's plot, or scatter plot. And finally, you can base the rest of your analysis on your transformed features, rather than the original 100 feature dataset.

PCA is a very fast algorithm and helps you vaporizes redundant features, so when you have a high dimensionality dataset, start by running PCA on it and then visualizing it. This will better help you understand your data before continuing.

Projecting a Shadow

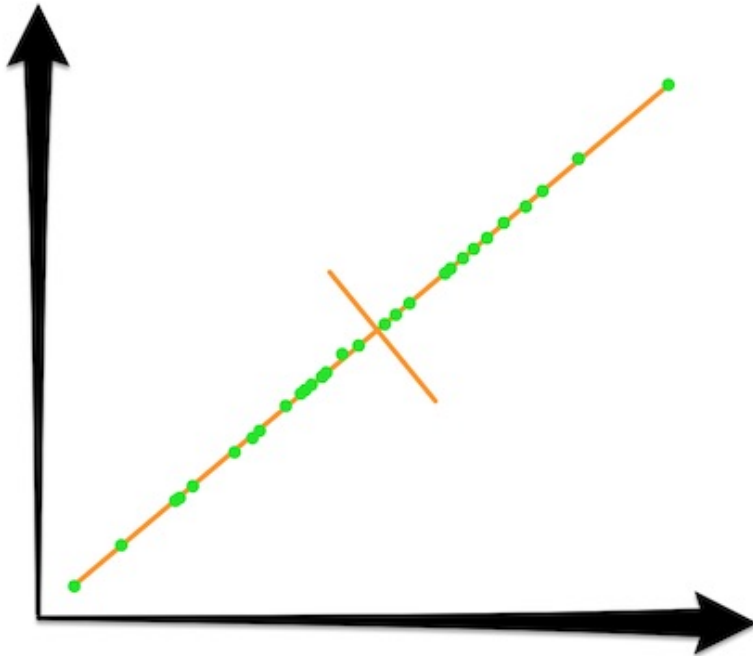
By transforming your samples into the feature space created by discarding under-prioritized features, a lower dimensional representation of your data, also known as *shadow* or *projection* is formed. In the shadow, some information has been lost—it has fewer features after all. You can actually visualize how much information has been lost by taking each sample and moving it to the nearest spot on the projection feature space. In the following 2D dataset, the orange line represents the principal component direction, and the gray line represents the second principal component. The one that's going to get dropped:



By dropping the gray component above, the goal is to project the 2D points onto 1D space. Move the original 2D samples to their closest spot on the line:




Once you've projected all samples to their closest spot on the major principal component, a *shadow*, or lower dimensional representation has been formed:



The summed distances traveled by all moved samples is equal to the total information lost by the projection. In an ideal situation, this lost information should be dominated by highly redundant features and random noise.



 English ▾

© 2012-2017 edX Inc. All rights reserved except where noted. EdX, Open edX and the edX and Open EdX logos are registered trademarks or trademarks of edX Inc. | 粤ICP备17044299号-2

POWERED BY
OPENedX®

