Unsupervised Learning, Part 1

Dimensionality Reduction and PCA
Introduction to Statistical Learning, Chapter 10

Recall: Two Main Types of Machine Learning

Supervised Learning: Given *labeled* data, find a function that is able to predict the label from the input data. Have an *outcome* we want to predict from the *features*.

Unsupervised Learning: Given *unlabeled* data, find underlying structure in the dataset. *Features* but *no known outcome*.

Types of Unsupervised Learning

Four main goals:

- **Embedding:** Find a lower-dimensional representation of the data, without losing too much “resolution”
- **Clustering:** discover “clumps” of points (eg. customer segmentation)
- **Density Estimation:** Approximate the probability distribution of the data (think KDE from seaborn).
- Finding good explanations (**hidden causes** or **sources**) of the data (model the data generation process)

Unsupervised Learning Challenges

- No simple goal, like with supervised learning.
- No teacher/supervisor to provide “correct answers”.
- Hard to assess the results obtained - no labels, so we can’t compute accuracy or MAE.

Dimensionality Reduction

Dimensionality Reduction

Goal: Find low-dimensional representation of our high-dimensional data

But do it in a way that preserves the “local” and “global” structure of the dataset, so that “near” points stay “near” and “far” points stay “far” (preserve the variance).

Uses:

- visualization (plots of >2 dimensions are hard)
- simpler representation of the data.

Principal Components Analysis (PCA)

Finds a low-dimensional representation of a data set that contains as much as possible of the variation.

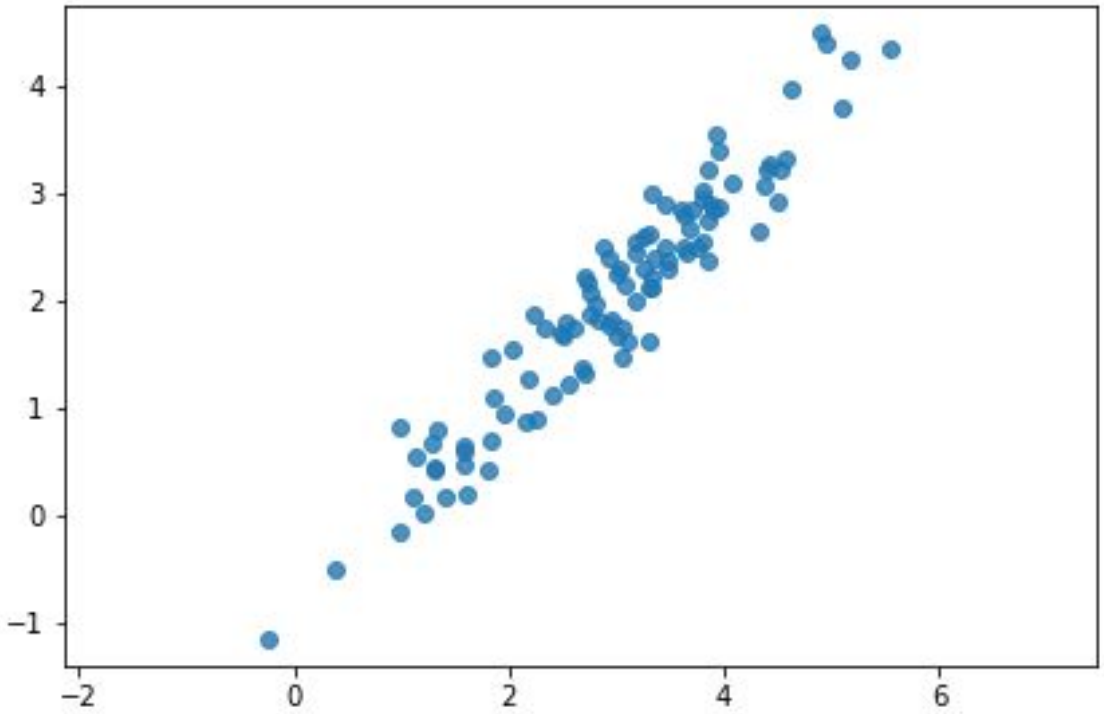
Useful when working with a high-dimensional dataset, where not all of the dimensions are particularly “interesting”.

A **linear** method, so if you have particularly weird, nonlinear structure in your dataset, it will not be able to detect it.

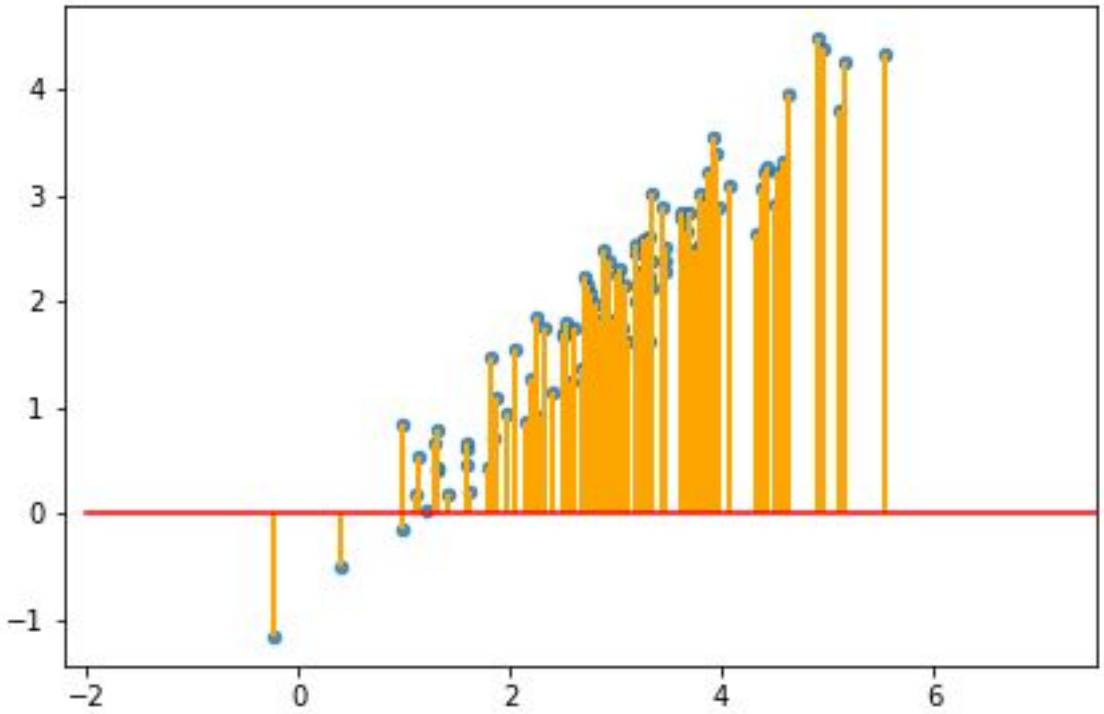
Principal Components Analysis (PCA)

The following slides will show an example of doing a one-dimensional projection of a two-dimensional dataset.

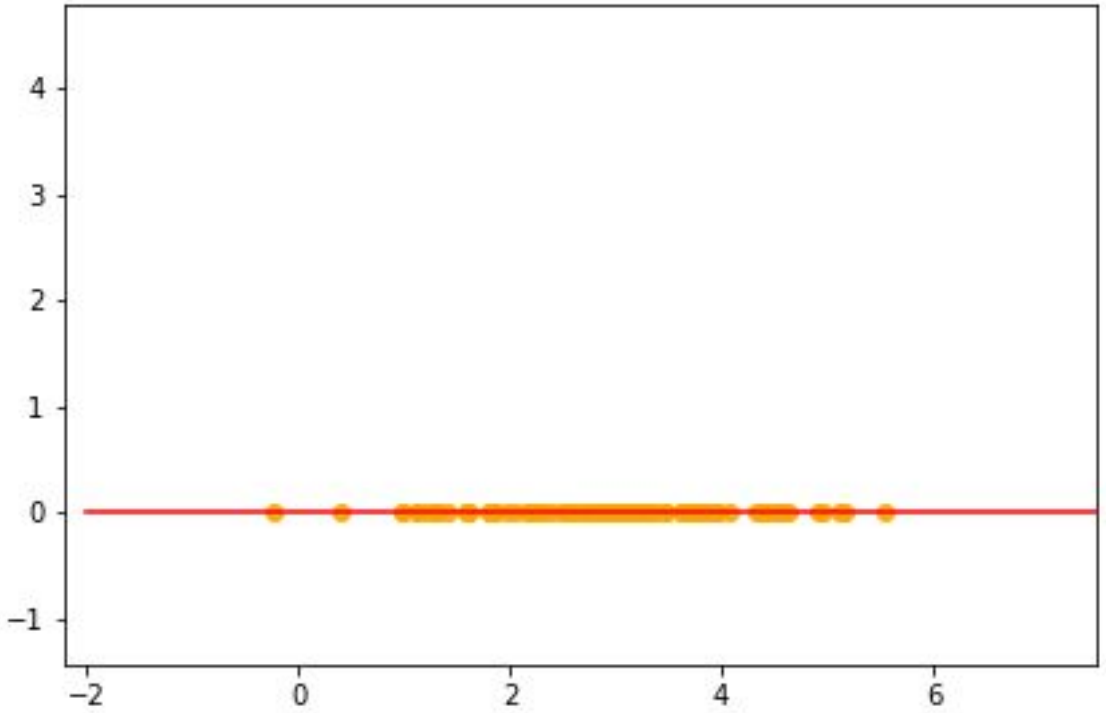
In reality, PCA is more commonly used to project 3-or-higher-dimensional datasets onto 2 dimensions, so you'll have to use your imagination.



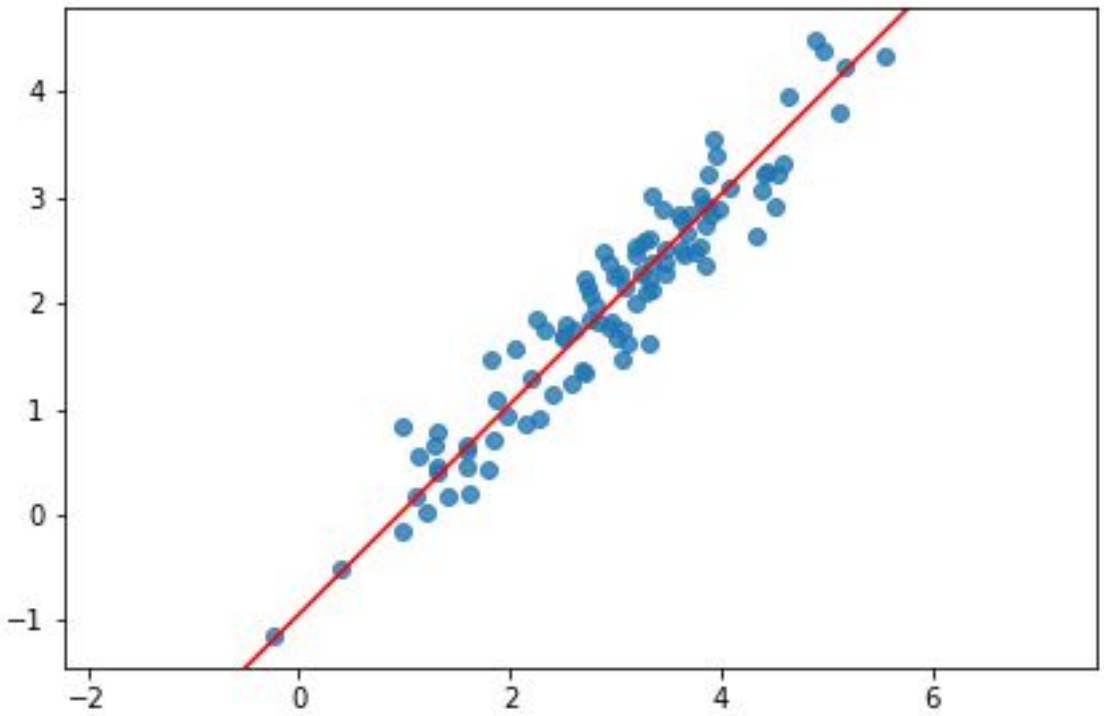
Given this 2-dimensional dataset, can we project it onto a 1-dimensional subset (line) so that we preserve most of the variance (information)?



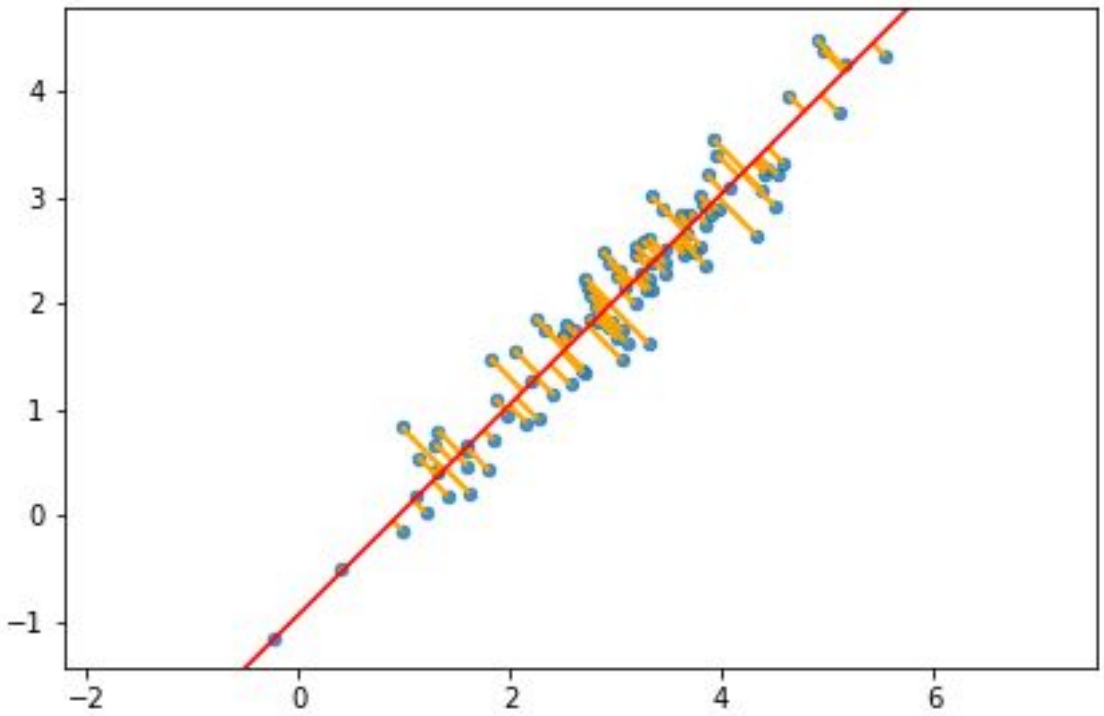
One option is to project onto one of the coordinate axes, say, the *x*-axis. This effectively “forgets” about one of the variables.



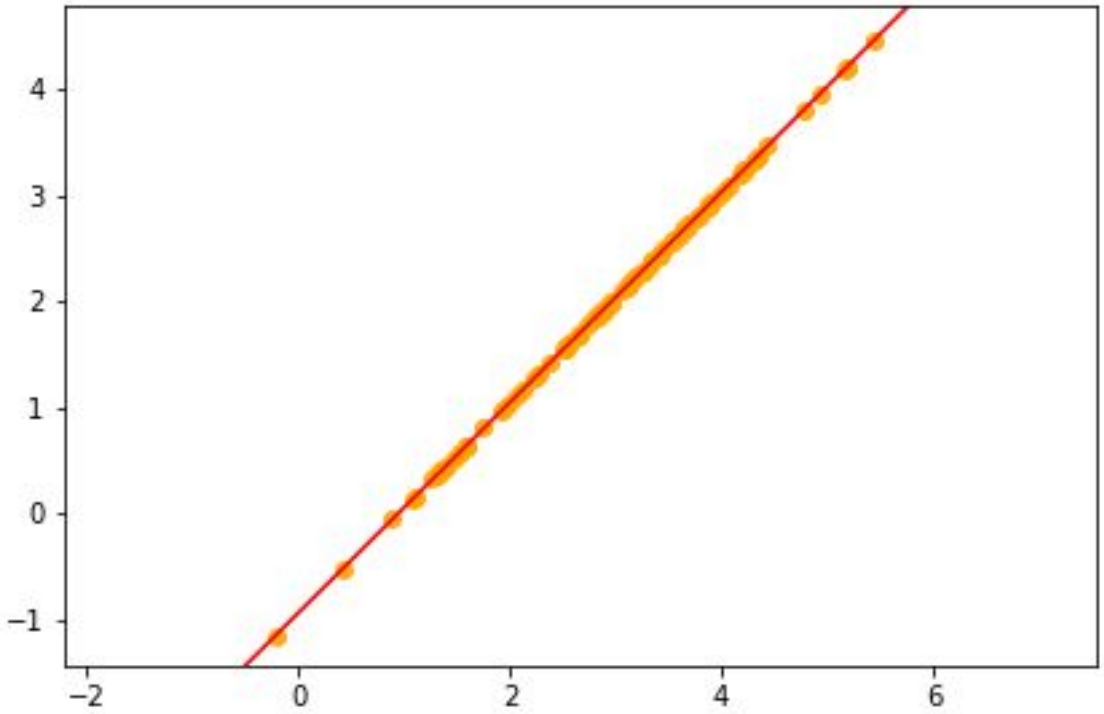
Can we do better than this, though?



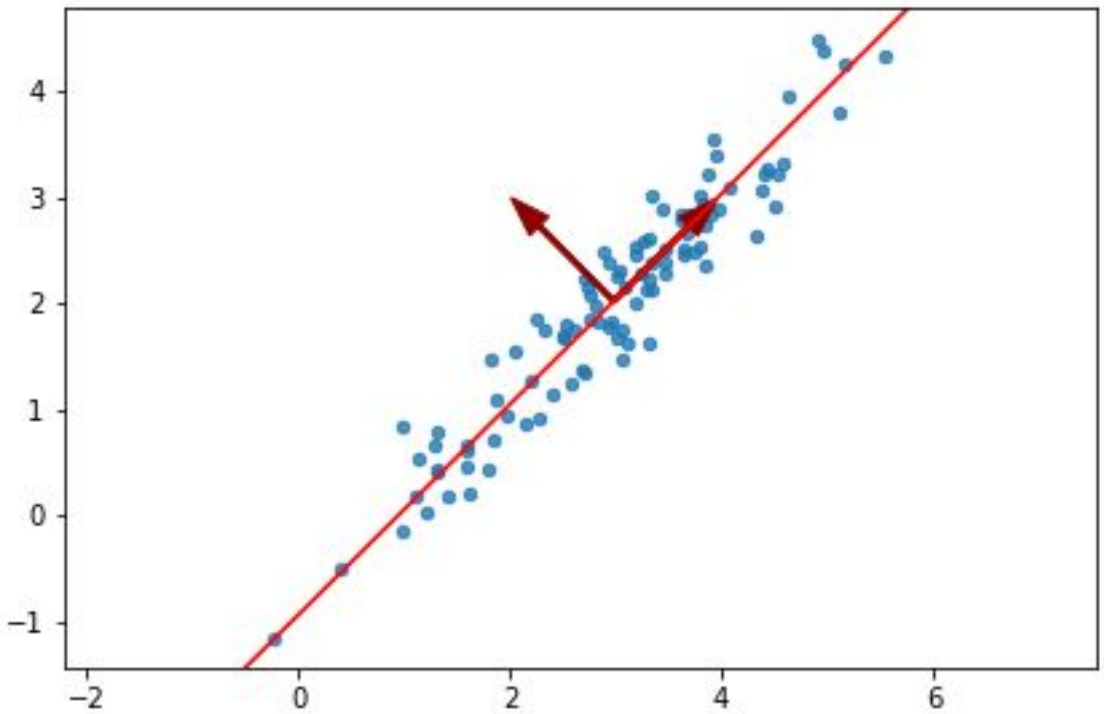
In this (extreme) example, there is a diagonal line that stands out.



Now project each point onto this line.
PCA finds the line that makes these projection lengths as small as possible (sum of square distances)



This give us a one-dimensional representation of our dataset (the distance along this line)



PCA identifies the most important directions (vectors). The vector along the line is the first principal component, and the other is the second principal component.

Principal Components Analysis (PCA)

Does not necessarily *remove* variables, but instead combines the original variables into linear combinations to create new variables (blends of the original variables).

Once we find our principal components, we can project onto the subspace generated by them to get a lower-dimensional representation of our original dataset.

Note: variables should be standardized (centered + scaled) before performing PCA (but most PCA libraries do this by default)

Principal Components Analysis (PCA)

How is it done?
Linear algebra (eigendecomposition of covariance matrix) 😦

The proportion of variance explained can be used to determine how much variance is retained/explained by the chosen principal components.

Example Notebook

PCA.Rmd