Dimensional Reduction with PCA

Kris Gunsalus

9/26/2018

Contents

Clustering or Dimensional Reduction: When and Why?

Which genes show the most similar expression patterns? Which conditions have the most similar "state" in terms of their expressed genes? How many distinct "classes" are there? => **Clustering**

How can I summarize the entire dataset in terms of the distribution of the data? Can I represent the data with a smaller number of variables / descriptors? How can I visualize the relationships between different conditions? => **Dimensional reduction**

Motivation

Often, the amount of "information" present in a dataset – in terms of the total variability – is not uniformly distributed across measurements. Why? If two dimensions (experimental conditions) are highly correlated, then there is some dependency between them, and amount of information gained by including the second dimension in your analysis is small. In such cases, we can *project* the data onto a smaller number of dimensions. **Figure 1** shows a simple 2D example.

Both PCA and tSNE seek to simplify the way data are represented while preserving the large-scale structure of the data, i.e the relationships among informative features.

PCA is used for a wide range of applications, including image analysis, data compression, feature extraction, and data visualization. tSNE is a more recent method that uses a different philosophy to accomplish similar goals. While PCA seeks to explain most of the variation, tSNE focuses more on local neighborhoods. You can broadly think of these as different ways to look at relationships, based on "far"-ness vs. closeness.

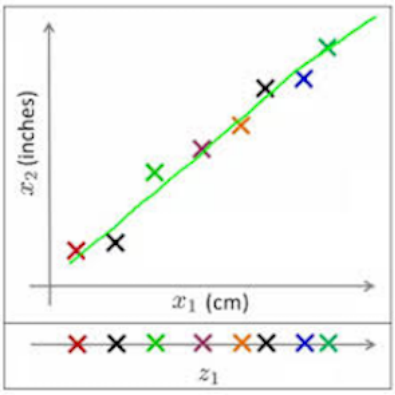

Figure 1: Projecting data in two dimensions onto a single dimension

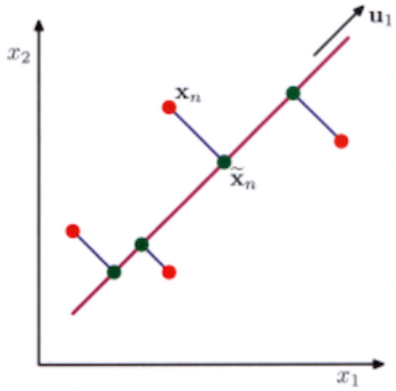

Figure 2: Projection onto a principal subspace

Principal Components Analysis (PCA)

Overview

Here our goal is to identify large-scale patterns across the data. Principal components enable us to summarize data using a smaller number of variables that collectively capture most of the variability, in the original dataset. Consider the following two-dimensional dataset (**Figure 2**).

The basic idea is that we want to *change coordinate systems* – from x_1 and x_2 to $\mathbf{u_1}$ and something else – so that the **maximum amount of variation is distributed along orthogonal axes**. To do this, we make a *linear projection* of the data onto a lower-dimensional subspace, which we call the *principal subspace*. In this new subspace, $\mathbf{u_1}$ is the "first principal component", and most of the variation in the data is distributed along this axis. In other words, it is the dimension along which the data are most "spread out".

There are two equivalent ways to think about PCA:

- **Maximize the variance** of the projected data (in Figure 1, along the $\mathbf{u_1}$ axis)
- **Minimize the average projection cost**, i.e. the mean squared distance between data points and their projections (in Figure 1, the distance from x_n to to $\tilde{x}$)

The two formulations give rise to the same algorithm.

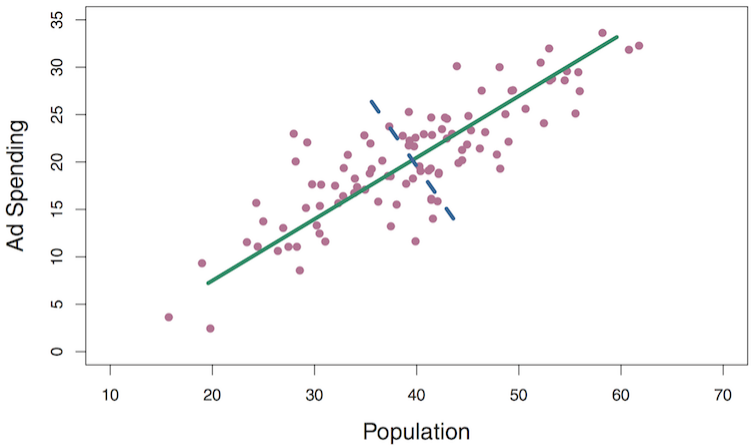

FIGURE 6.14. *The population size (*pop*) and ad spending (*ad*) for* 100 *different cities are shown as purple circles. The green solid line indicates the first principal component, and the blue dashed line indicates the second principal component.*

Figure 3: Population vs. Ad Spending in 100 US cities

A simple case: PCA in two dimensions

Now let's look at a real-life example, from Chapter 6.3.1 of *Introduction to Statistical Learning* (**Figure 3**).

We see from the figure that most of the variation in the data falls along the green line. This is the first principal component (PC1). If we were to remove this variation, then we could describe the rest of the variation in terms of the orthogonal dimension shown by the dashed blue line.

Now we have effectively changed our coordinate system from "Population" (x) vs. "Ad Spending" (y) to a **new coordinate system**, PC1 and PC2 (**Figure 4**). Notice that we have transformed the data by *mean centering*, so that we represent each original dimension as a deviation from its mean. We also *scaled* the data so that the standard deviation of each variable is 1.

Relationship between original and PC coordinate systems

Now we want a way to quantify the **loadings** of the original variables on to our new dimensions. Let's define a set of M principal components, and call them $Z_1 ... Z_m$. For the above example, we can now write, for PC1:

$$Z_1 = 0.839 * (pop - \overline{pop}) + 0.544 * (ad - \overline{ad})$$

Now something interesting has happened. We have transformed our original set of p predictors (population, ad spending) of n variables (cities) to a new set of predictors in M dimensions: the **principal components**.

In this example, that doesn't seem like a big deal, but consider gene expression studies, where we measure thousands of genes (our n variables) across many samples (call this p). We can describe this as an n x p matrix.

By performing PCA, we can reduce the dimensionality of our dataset from p to something much smaller that explains most of the variation in the dataset. This is particularly convenient for large datasets, such as *single-cell studies*, since our M principal components will be some **linear combination** of our original measurements, where $M < p$.

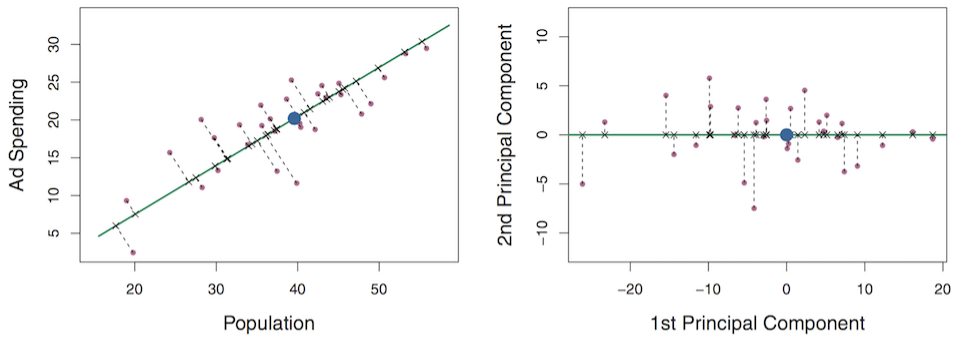

FIGURE 6.15. *A subset of the advertising data. The mean* pop *and* ad *budgets are indicated with a blue circle.* Left: *The first principal component direction is shown in green. It is the dimension along which the data vary the most, and it also defines the line that is closest to all n of the observations. The distances from each observation to the principal component are represented using the black dashed line segments. The blue dot represents* $(\overline{\texttt{pop}}, \overline{\texttt{ad}})$. Right: *The left-hand panel has been rotated so that the first principal component direction coincides with the x-axis.*

Figure 4: Data transformation to PC1 and PC2

A 3D example

What happens when we have more than just two features to compare? Let's consider a small dataset simulated in three dimensions (**Figure 5**). Now we see that, in the same way that PC1 captures the most variation in the dataset, together the first M principal components provide the best M-dimensional approximation of the original dataset. We can extend this approach to any number of dimensions.

As p increases, it rapidly becomes unweildy to visualize how the data are distributed by plotting every pairwise combination of features (since there are $p * (p-1)/2$ pairwise combinations). Instead, PC plots provide a nice way to visualize the most significant variation across the dataset. Looking at 2D plots of the first few PCs usually provides a pretty good summary of the overall variation in the original data.

Mathematical formulation for PCA

Let's now find a mathematical way to describe what we have done that generalizes to multiple dimensions. Let's call our set of p predictors $X_1, X_2 ... X_p$, and M represent $M < p$ linear combinations of them. Note that for each feature (condition) X we have n observations (genes), so that each X is a vector of length n. For PC1, we can now write:

$$Z_1 = \phi_{11} X_1 + \phi_{21} X_2 + ... + \phi_{p1} X_p$$

We say that Z_1 is the *first principal component* of the set of features X. It is a *normalized linear combination* of these that has the *largest variance* across the p predictors.

Loading vectors

We call the coefficients ϕ the **loadings** of the predictors onto the principal components. We write the loading vector for the first principal component as:

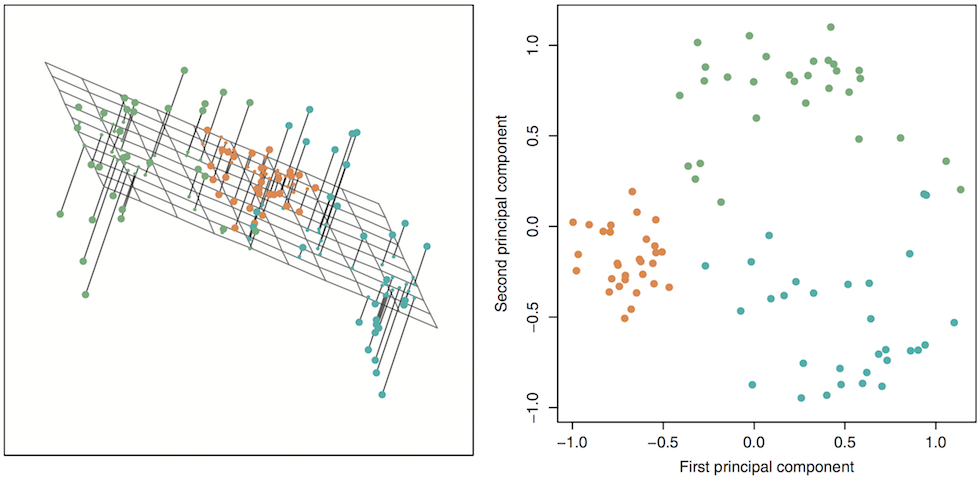

FIGURE 10.2. *Ninety observations simulated in three dimensions.* Left: *the first two principal component directions span the plane that best fits the data. It minimizes the sum of squared distances from each point to the plane.* Right: *the first two principal component score vectors give the coordinates of the projection of the 90 observations onto the plane. The variance in the plane is maximized.*

Figure 5: PC1 and PC2 for a 3D dataset

$$\phi_1 = (\phi_{11}, \phi_{12}, ..., \phi_{p1})$$

In the above example, the loading of "Population" onto the first principal component was 0.839, and the loading of "Ad spending" onto PC1 was 0.544.

Normalization

Notice that we also said the the linear equations are *normalized*. This is because we want the *total variance* for $Z1$ across the entire dataset to sum to 1, that is:

$$\sum_{j=1}^{p} \phi_{j1}^2 = 1$$

For the advertising example above, we can see that $\theta_{11}^2 + \theta_{21}^2 = (0.839)^2 + (0.544)^2 = 1$.

Normalization allows us to quantify the proportion of the total variance that is distributed along each PC. When $M = p$, we have explained 100% of the variation in the original dataset.

Scaling

In addition, in order to make our measurements across the different variables (samples) more comparable, we *center* the data points in each sample by subtracting each measurement from the sample mean, and then *scale* the total variation in each sample so that the standard deviation equals one: $s = 1$. (Notice that this is reflected in the units shown for PC1 and PC2 in Figure 3.)

Scaling is much more important when variables are measured in different units, such as the population of cities vs. the amount of ad money spent in each state.

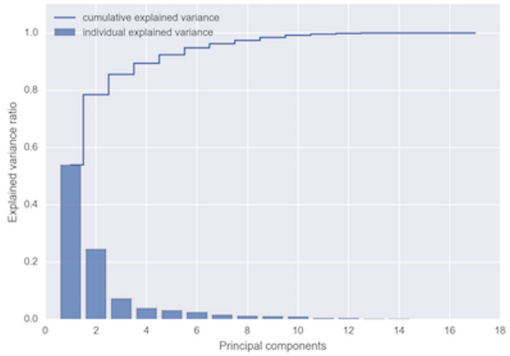

Figure 6: Scree plot for 4 principal components

PC scores

To complete our formal description, we need a way to describe the relationship between each PC and each of the original observations x_{ij}, where $i = 1...n$ and $j = 1...p$. For an n x p matrix of expression data, for example, we index each gene with i and each sample with j.

Each of the M principal components Z comprises n linear combinations of p terms. We call these *scores* and denote them as $z_{11}, ..., z_{n1}$. The scores for PC1 take the form:

$$z_{i1} = \phi_{11}x_{i1} + \phi_{21}x_{i2} + ... + \phi_{p1}x_{ip}$$

Just as the loadings relate each PC to the original set of features $X_1...X_p$, so the scores for each PC relate to the individual elements x_{ij}.

How to choose the right M?

In practice, since we are interested in simplifying our description of the data, we want M to be smaller than p. There is no single correct answer to choosing M. One method that is commonly used is to "eyeball" the data using a scree plot (**Figure 6**).

A common approach is to look for an "elbow" in the plot, and choose the number of PCs that is one less (because at the elbow, adding PCs adds less and less to the total explained variation). In this example, 2 principal components seem sufficient to explain most of the data. Often, you will have more than 4 PC.

Another way is to decide on the total proportion of the variance you want to be able to explain. For example, you could choose a cutoff of 80-95%, depending on your application.

Additional Reading

- ISLR (Introduction to Statistial Learning), Chapter 6.3 (PDF provided on website with class notes)