Chapter 10: Unsupervised Learning

Credit: https://xkcd.com/1425/

This chapter will focus on methods intended for the setting in which we only have a set of features $X_1, \dots, X_p$ measured on n observations.

We are not interested in prediction because we have no response Y.

Goal: discover interesting things about the measurements $X_1, \dots, X_p$

- Is there an informative way to plot the data?
- Can we discover subgroups among variables or observations?

1 The Challenge of Unsupervised Learning

Supervised learning is a well-understood area.

You now have a good grasp of supervised learning.

If you are asked to predict a binary response you have many well developed tools at your disposal:

logistic regression, bagged trees, boosted trees, LDA, RF, SVM, etc.

and have a clear understanding of how to assess quality of your results:

cross-validation, validation on an independent test set

↳ LOO, k-fold, etc.

In contrast, unsupervised learning is often much more challenging.

more subjective, no single goal for the analysis, e.g. prediction.

Unsupervised learning is often performed as part of an *exploratory data analysis*.

1st part of analysis before models are fit.

It can be hard to assess the results obtained from unsupervised learning methods.

No universally accepted mechanism for performing cross-validation or validation on a test set

Because there is no way to "check our work" with response variable

⟶ we don't know the true answer!

Techniques for unsupervised learning are of growing importance in a number of fields.

Cancer research: assay gene expression levels in 100 patients and look for subgroups among cancer samples to better understand the disease.

Online shopping: identify similar groups of shoppers and show preferential items that they may be particularly interested in.

My research — Entity resolution: Many noisy databases without unique identifying attributes ⟶ can we find the matches or links?

2 Principal Components Analysis

We have already seen principal components as a method for dimension reduction.

When faced with a large set of correlated variables, we use principal components to summarise with a smaller number of "representative" variables that collectively explain most of the variability in our original dataset.

PC directions = directions in feature space along which original data are highly variable.
↳ define lines and subspaces that are as close as possible to the data cloud.

PCR = use principal components as predictors in a regression model instead of original variables.

Principal Components Analysis (PCA) refers to the process by which principal components are computed and the subsequent use of these components to understand the data.

Unsupervised approach (involves only features $X_1, \dots, X_p$, no response Y).

Apart from producing derived variables for use in supervised learning, PCA also serves as a tool for data visualization.

Visualizing observations or of variables.

2.1 What are Principal Components?

$X_1, \dots, X_p$

Suppose we wish to visualize n observations with measurements on a set of p features as part of an exploratory data analysis.

We could do this by examining 2D scatterplots of the data which contain n observations on 2 features.

$\Rightarrow \binom{p}{2} = \frac{p(p-1)}{2}$ scatter plots, e.g. w/ $p = 10 \Rightarrow 45$ plots.

- Too many to look at.
- likely no plot will be informative because they only contain a small fraction of information in our data.

For visualization in high dimensions.

Goal: We would like to find a low-dimensional representation of the data that captures as much of the information as possible.

Then plot observations in lower dimensional space.

PCA provides us a tool to do just this.

It finds low-dimensional representation of a data set that contains as much as possible of the variation (information).

Idea: Each of the n observations lives in p dimensional space, but not all of these dimensions are equally interesting.

PCA seeks a small number of dimensions that are as interesting as possible.

"interesting" = amount observations vary along each dimension.

Each dimension found in PCA is a linear combination of p features.

The *first principal component* of a set of features $X_1, \dots, X_p$ is the normalized linear combination of the features

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

normalized: $\sum_{j=1}^{p} \phi_{j1}^2 = 1$ (otherwise we could result in arbitrarily large variance).

$\phi_{11}, \dots, \phi_{p1}$ are called "loadings" of first principal component $\phi_1 = (\phi_{11}, \dots, \phi_{p1})^T$ "loading vector"

that has the largest variance.

Given a $\boldsymbol{n} \times \boldsymbol{p}$ data set $\boldsymbol{X}$, how do we compute the first principal component?

1. Assume each variable has been centered (i.e. each column has mean zero) – only care about variances.
2. look for linear combination of the form

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

w/ largest variance, subject to

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

i.e. solve the following optimization problem:

$$\underset{\phi_{11}, \dots, \phi_{p1}}{\text{maximize}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j1} x_{ij} \right)^2 \right\} \text{ subject to } \sum_{j=1}^{p} \phi_{j1}^2 = 1.$$

can write this way b/c columns are centered

$$\Rightarrow \frac{1}{n} \sum_{i=1}^{n} x_{ij} = 0 \Rightarrow \frac{1}{n} \sum_{i=1}^{n} z_{i1} = 0$$

so above is variance of $z_{i1}, i = 1, \dots, n$.

Solved using eigen decomposition (beyond scope of this class).

$z_{11}, \dots, z_{1n}$ are called "scores" of the first principal component.

There is a nice geometric interpretation for the first principal component.

The loading vector ϕ_1 defines the direction in the feature space along which the data vary the most

If we project n data points onto this direction we get the scores $z_{11}, \dots, z_{n1}$.

After the first principal component Z_1 of the features has been determined, we can find the second principal component, Z_2. The second principal component is the linear combination of $X_1, \dots, X_p$ that has maximal variance out of all linear combinations that are uncorrelated with Z_1.

The second principal component scores are

$$z_{i2} = \phi_{12} x_{i1} + \dots + \phi_{p2} x_{ip}$$

ϕ_2 = second principal component loading vector

Z_2 uncorrelated w/ Z_1

$\Longleftrightarrow$

ϕ_2 orthogonal to ϕ_1

$p = 2$

in 2D space, there is only one possibility for ϕ_2

But $p > 2$ there are multiple options orthogonal.

To find Z_2, solve a similar optimization problem w/ additional constraint:

$$\underset{\phi_{21}, \dots, \phi_{2p}}{\text{maximize}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j2} x_{ij} \right)^2 \right\}$$

subject to $\sum_{j=1}^{p} \phi_{j2}^2 = 1$ and ϕ_2 orthogonal to ϕ_1 $\left(\sum_{j=1}^{p} \phi_{j2} \phi_{j1} = 0 \right)$.

Once we have computed the principal components, we can plot them against each other to produce low-dimensional views of the data.

each of the 50 states, # arrests per 100,000 residents for each of 3 crimes

```
str(USArrests)
```

```
## 'data.frame':    50 obs. of  4 variables:
##  $ Murder  : num  13.2 10 8.1 8.8 9 7.9 3.3 5.9 15.4 17.4 ...
##  $ Assault : int  236 263 294 190 276 204 110 238 335 211 ...
##  $ UrbanPop: int  58 48 80 50 91 78 77 72 80 60 ...
##  $ Rape    : num  21.2 44.5 31 19.5 40.6 38.7 11.1 15.8 31.9 25.8 ...
```

% population in state living in an urban area.

```
pca <- prcomp(USArrests, center = TRUE, scale = TRUE) # get loadings

summary(pca) # summary
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4
## Standard deviation     1.5749 0.9949 0.59713 0.41645
## Proportion of Variance 0.6201 0.2474 0.08914 0.04336
## Cumulative Proportion  0.6201 0.8675 0.95664 1.00000
```

PVE

First two principal components explain 86.75% of variability in the data
last two only 13% ⇒ looking at first 2 is good summary.

```
pca$rotation # principal components loading matrix
```

ϕ_1 ϕ_2 ϕ_3 ϕ_4

```
##                PC1        PC2        PC3         PC4
## Murder   -0.5358995  0.4181809 -0.3412327  0.64922780
## Assault  -0.5831836  0.1879856 -0.2681484 -0.74340748
## UrbanPop -0.2781909 -0.8728062 -0.3780158  0.13387773
## Rape     -0.5434321 -0.1673186  0.8177779  0.08902432
```

```
## plot scores + directions
biplot(pca)
```

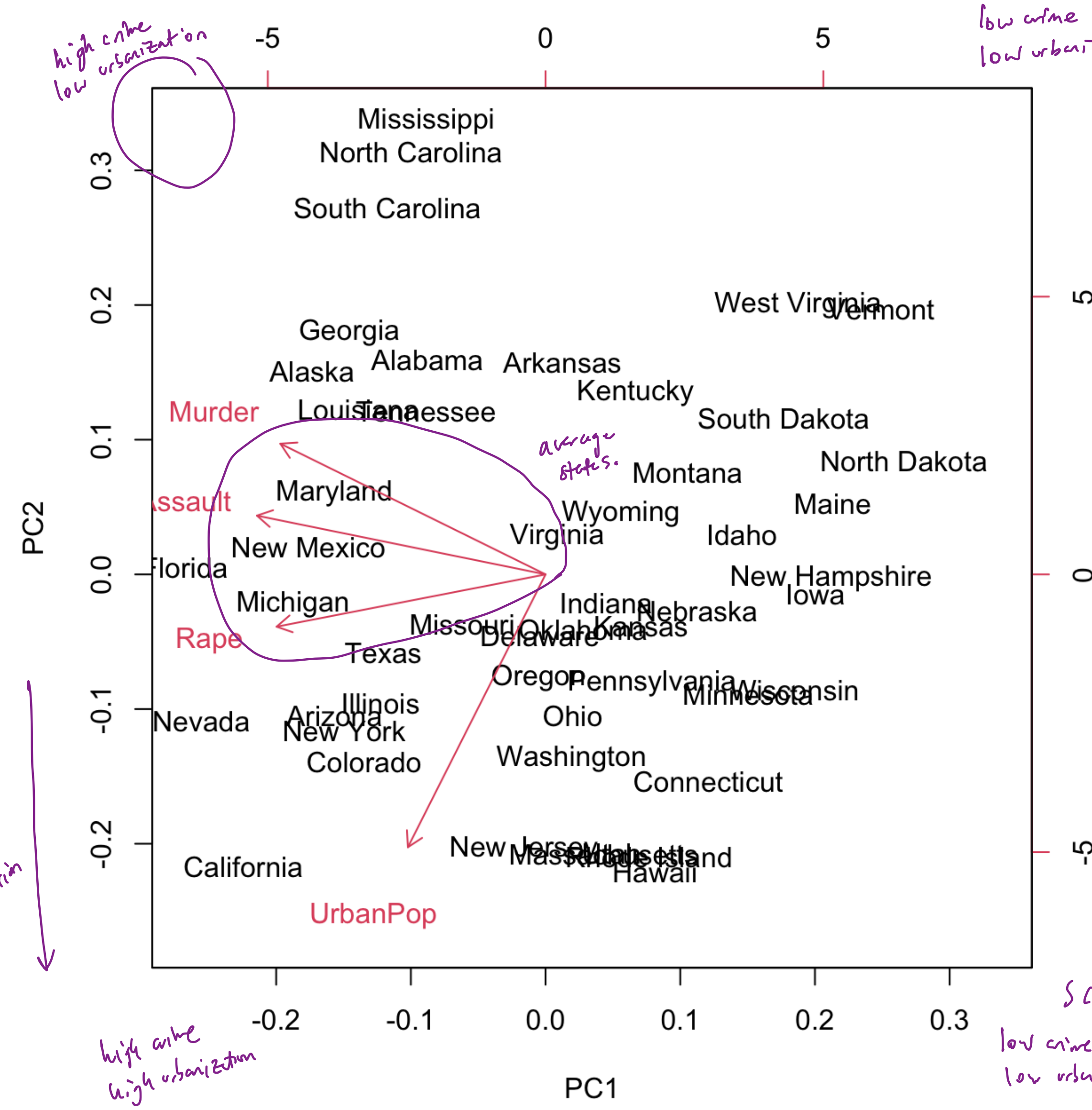
more crime.

high crime
low urbanization

low crime
low urbanization

Crime related variables are correlated.

more urbanization



high crime
high urbanization

Scores.

low crime
low urbanization.

First loading places approximately equal weight on 3 crimes and less weight on Urban pop.

$\Rightarrow$ this component $\approx$ measure of serious crimes

Second loading places most weight on Urban pop $\Rightarrow$ $\approx$ level of urbanization in a state.

2.2 Scaling Variables

We've already talked about how when PCA is performed, the varriables should be centered to have mean zero.

Also the results depend on whether variable have been individually scaled to have same sd.

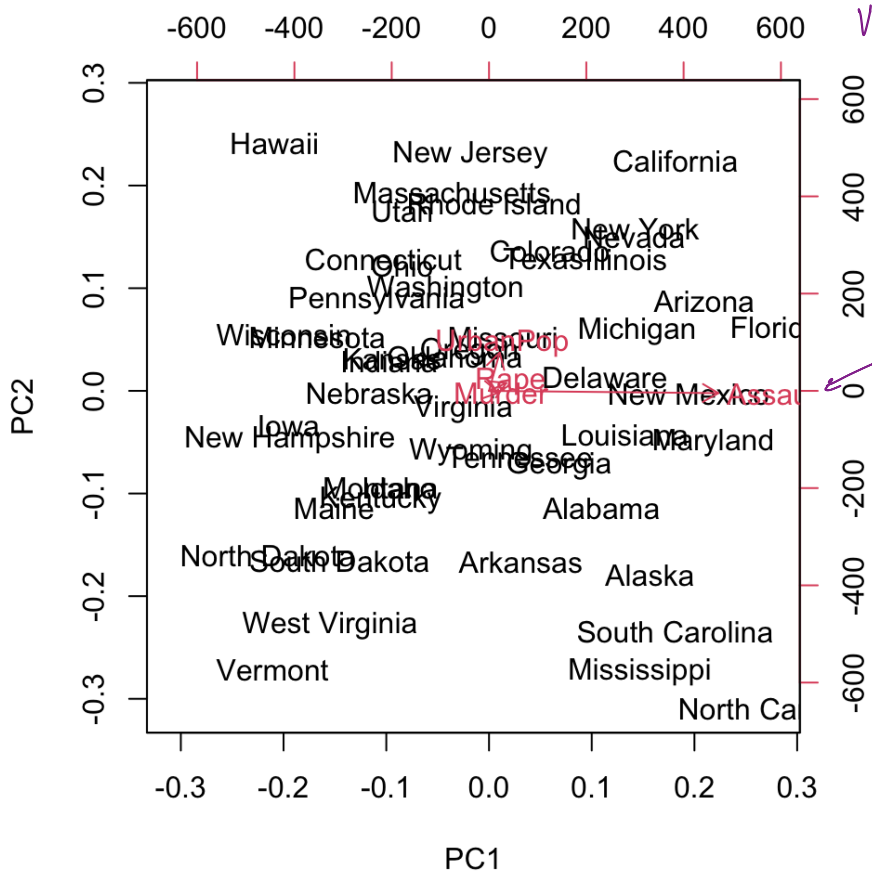
This is in contrast to other methods we've seen before.

e.g. linear regression when we multiply a variable by c the corresponding coefficient is change by a factor of $\frac{1}{c}$.

same data as before, didn't scale. Variables are measured in different units

Crime: #/100,000 people

Urbanpop: percentage



large loading on assault for 1st PC (large variance due to its scale).

Undesirable for PCA to depend on something as arbitrary as scale ⇒ Scale each variable to have st. dev = 1.

UNLESS: all variables are measured on same units ⇒ might not want to scale them.

2.3 Uniqueness

Each principal component loading vector is unique, up to a sign flip.

$\Rightarrow$ different software should result in same prin. component loading vectors, but sign might flip.

Signs may differ because each principal component loading specifies a direction in p-space $\downarrow$ a line that extends in either direction

Flipping the sign has no effect since the direction doesn't change.

Similarly, the score vectors are unique up to a sign flip.

$$Var(Z) = Var(-Z).$$

2.4 Proportion of Variance Explained

We have seen using the `USArrests` data that we can summarize 50 observations in 4 dimensions using just the first two principal component score vectors and the first two principal component vectors.

Question:

How much of the information ($\rightarrow$ variability explained.) in a given data set is lost by projecting the observations on to the first two principal component vectors?

More generally, we are interested in knowing the *proportion of variance explained (PVE)* by each principal component.

Total variance in data set: $\displaystyle\sum_{j=1}^{p} Var(X_j) = \sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2$

Variance explained by m^{th} principal component: $\displaystyle\frac{1}{n}\sum_{i=1}^{n} z_{im}^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{jm}x_{ij}\right)^2$

$\Rightarrow$ PVE by m^{th} principal component: $\displaystyle\frac{\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{jm}x_{ij}\right)^2}{\sum_{j=1}^{p}\sum_{i=1}^{n}x_{ij}^2}$ (positive quantity).

cumulative PVE for 1^{st} M components: sum PVE first M

2.5 How Many Principal Components to Use

In general, a ~~*ntimesp*~~ $n \times p$ matrix $\boldsymbol{X}$ has $\min(n-1, p)$ distinct principal components.

We are probably not interested in all of them.

Rather, we would like to just use the first few principal components in order to visualize or interpret the data.

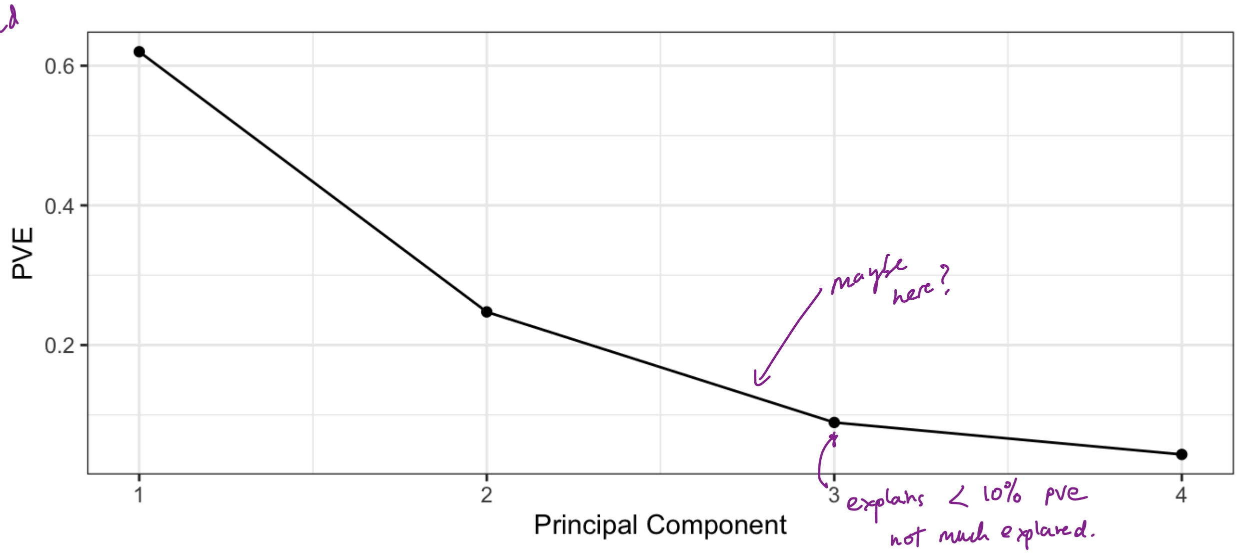
Want to use smallest # required to get a good understanding of structure of data.

How many?

No one simple answer.

We typically decide on the number of principal components required by examining a *scree plot*.

sometimes called "elbow" plot.



maybe here?

explains < 10% PVE
not much explained.

look for a point that has an "elbow", where plot stops dropping so sharply

This is ad hoc, but the question of how many is "enough" is not well defined.
depends on problem, the data, your goals.

Unsupervised EDA

Usually plot first two components look for "interesting" patterns. If there are none, probably won't be interesting later components.

If first 2 are interesting, keep looking!

For supervised PCR → there is a good way to choose # components: CV!

2.6 Other Uses for Principal Components

PCR

We've seen previously that we can perform regression using the principal component score vectors as features for dimension reduction.

Many statistical techniques can be easily adapted to use the $n \times M$ matrix whose columns are the first $M << p$ principal components. instead of full $n \times p$ data set X

e.g.: other types of regression, classification, clustering.

This can lead to *less noisy* results.

Since usually signal is concentrated in its first few principal components.