Week 3 - Principal component analysis

CM of 1h30 followed by a TP of 1h30

We start by wrapping last lecture

-- **Model selection**

Now that you know how to get an **estimate of the generalization error** of a model from data, we can talk about model selection in more precise terms.

Suppose that we were given a dataset with p predictors and we want to know whether a **linear model using only a subset** of them should be preferred.

> Figure with example on the `mtcars` dataset

There are mainly **three strategies** for reducing the number of parameters of a model:

- *Subset selection*: we **identity a subset** of the p predictors that we think is the most related to the response and then fit a linear modeal with least squares to it
- *Shrinkage*: we fit a model to all p predictors but using an extended loss function for the least-squares problem of fitting the parameters. This extension is called a **regularizer** and aims to reduce to zero the parameters of the predictors that are not important for describing the response (we talked a bit about it last week on TD)
- *Dimension reduction*: this approach involves **projecting** the p predictors to a lower-dimensional space defined by linear combinations of variables. A very **well known method** to do this kind of thing is the principal component analysis (aka **PCA**) and we will talk about it later today.

The **simplest way** of doing subset selection is to **consider all 2^p possible models** with subsets of the predictors and then choose the one which has the smallest cross-validated prediction error.

- The problem with this strategy is that for $p > 10$ things start to become expensive (1024 models to evaluate) to calculate and for $p > 30$ simply untractable (more than 1 billion models to evaluate).

A **more appealing** method for subset selection searches for subsets by doing a **forward stepwise selection**, where we incrementally augment the linear model using the coefficients from a previous step

The procedure goes as follows:

1. Let $\mathcal{M}_0$ denote a model with no predictors (i.e. just the intercept)
2. For $k = 0, \ldots, p-1$ we do:

   (a) Consider all $p-k$ models that augment the predictors in $\mathcal{M}_k$ with one extra predictor

   (b) Choose the **best** amont these $p-k$ models and call it $\mathcal{M}_{k+1}$
   NB: Best is defined as having the **smallest training** or testing error. How so?
3. Select a single best model among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error
   NB: Why here we really **should not** compare the models in terms of training error?

This **amounts** to $1 + p(p+1)/2$ models **to evaluate** which is much smaller than 2^p.

- With $p = 20$ we have only ~ 200 models to check, as compared to $\sim 10^6$ in the other method

NB: There exists other options to assess the model error instead of cross-validation, like **AIC and BIC**. They are usually **faster to calculate**. These are scores decrease as the RSS (training error) decreases but are compensated by the increase in the model complexity (i.e. the number of coefficients). We **won't be using** them in this course

> Figure with example of forward stepwise selection on `mtcars` dataset

-- **Unsupervised learning**

In our classes so far, we have always had a dataset $\{(x_i, y_i)\}_{i \leq n}$ where each observation x_i was composed by a certain number of **features** $x_i = [x_{i1}, \ldots, x_{ip}]$ and a **response** y_i measured on those predictors.

Our goal was then to learn how these **input** features related to the **output** responses.

Today, we will consider a setting where we only have a set of data points x_i and are not interested in doing prediction. Rather, our goal is to **discover interesting things** about the measurements.

In other words, we want to do **exploratory data analysis**.

- Is there an informative way of visualizing the data?

  > **Figure 1**: Data in high-dimensional and too many pair plots to show
  >
  > With p features, if we do pairwise scatterplots we end up with $\binom{n}{p}$ figures
  >
  > E.g. for $p = 10$ this means 45 plots. Too much information to look at !

- Are there subgroups among the predictors? (e.g. predictors that tell the same information)

  > **Figure 2**: Highly correlated variables that can be considered as just one
  >
  > Take the simulated dataset from TP1

There has been a lot of research on unsupervised learning and it is very **challenging topic** to work on.

We focus on a very well known method called "principal component analysis" or simply PCA.

-- **PCA**

Our dataset is denoted $\mathcal{X} = \{x_i\}_{i \leq n} \subset \mathbb{R}^p$

Goal: For each data point x_i find a new representation $z_i = f(x_i) \in \mathbb{R}^d$ where $d \ll p$

Criteria: The new set $\mathcal{Z} = \{z_i\}_{i \leq n}$ should **capture as much information** as possible from $\mathcal{X}$

- Intuition: **keep the directions** of the space where the data has the **most variability**
- Discard uninteresting directions where data is approximately equal

> **Figure 3**: Example of the ellipse in 2D and transforming it to a line
>
> We are considering the **same simulated dataset** from the Exercise 1 in the TP
>
> $$\begin{array}{rcl} X_1 & = & \varepsilon_1 \\ X_2 & = & 3X_1 + \varepsilon_2 \end{array} \Leftrightarrow \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \end{bmatrix} \sim \mathcal{N}(\mu, C) \quad \text{with} \quad \mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \text{and} \quad C = \begin{bmatrix} 1 & 3 \\ 3 & 10 \end{bmatrix} \sigma^2$$
>
> Show that the variability is associated to the notion of **variance** over the projected axis.
>
> Show the animated figure with different axis for projection

Method: In PCA, f is a **linear transformation** of the predictors.

Considering that $d = 1$, we may define the principal component as

$$Z_1 = f(X) = \phi_{11}X_1 + \phi_{21} + \cdots + \phi_{p1}X_p$$

where the ϕ_{j1} are the **loadings** for principal component Z_1.

We normalize $\sum_{j=1}^{p} \phi_{j1}^2 = 1$ since **otherwise** letting them grow indefinetly would lead to $\text{Var}(Z_1)$ growing indefinetly as well.

We consider the actual observed samples from dataset $\boldsymbol{X} = \begin{bmatrix} x_{11} & \dots & x_{1p} \\ \vdots & \vdots & \vdots \\ x_{n1} & \dots & x_{np} \end{bmatrix}$ and write their **linear combinations** as:

$$z_{i1} = \sum_{j=1}^{p} \phi_{j1} x_{ij} = x_i^T \phi_1$$

- Note that since the ϕ_{j1} are normalized, we can see the linear combination as a **projection** of the observed sample onto vector $\phi_1 = [\phi_{11} \dots \phi_{p1}]^T$

The ϕ_{j1} are then chosen to **maximize the total variance of the transformed/projected variables** via the following optimization procedure:

$$\underset{\phi_{11},\dots,\phi_{p1}}{\text{maximize}} \quad \frac{1}{n}\sum_{i=1}^{n}(z_{i1} - \bar{z}_1)^2 \quad \Leftrightarrow \quad \underset{\phi_{11},\dots,\phi_{p1}}{\text{maximize}} \quad \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{j1}\Big(x_{ij} - \tfrac{1}{n}\sum_{i=1}^{n} x_{ij}\Big)\right)^2 \quad \text{s.t.} \quad \|\phi_1\|^2 = 1$$

such that $\phi_1^T \phi_1 = 1$.

When doing PCA:

- we **always re-center** the data so to make the calculations and interpretations easier and cleaner.
- we **often standardize** the predictors since having one of them with a variance which is too large as compared to the others will bias the results towards it, i.e. the principal direction will end up having just that predictor, since it is the component with maximum variance too.

We have then

$$\underset{\phi_1,\dots,\phi_p}{\text{maximize}} \quad \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{j1}x_{ij}\right)^2 \quad \text{subject to} \quad \sum_{j=1}^{p}\phi_{j1}^2 = 1$$

The loss function can be **rewritten** as:

$$\ell(\phi) = \frac{1}{n}\sum_{i=1}^{n}(\phi_1^T x_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i^T \phi_1)^2 = \frac{1}{n}\|\boldsymbol{X}\phi\|^2 = \phi^T\left(\frac{1}{n}\boldsymbol{X}^T\boldsymbol{X}\right)\phi_1 = \phi_1^T \boldsymbol{C} \phi_1$$

Where $\boldsymbol{X} = \begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix} \in \mathbb{R}^{n \times p}$ is the data matrix and $\boldsymbol{C} = \dfrac{1}{n}\boldsymbol{X}^T\boldsymbol{X} \in \mathbb{R}^{p \times p}$ is the **covariance** of the data

So to solve the optimization problem, the **Lagrangian** takes the loss function and the equality constraint,

$$\mathcal{L}(\phi_1, \lambda) = \ell(\phi_1) + \lambda(1 - \phi_1^T\phi_1)$$

and taking its gradient with respect to ϕ_1 to zero we have:

$$\nabla_{\phi_1}\mathcal{L}(\phi_1, \lambda) = 2\boldsymbol{C}\phi_1 - 2\lambda\phi_1 = 0 \Leftrightarrow \boldsymbol{C}\phi_1 = \lambda\phi_1$$

which is just a problem involving **eigenvectors** and eigenvalues.

Note that $\boldsymbol{C}$ is **symmetric positive definite** meaning that:

- Its eigenvectors are **orthonormal**
- Its eigenvalues are all strictly larger than zero

$$\boldsymbol{C} = \boldsymbol{Q}\Lambda\boldsymbol{Q}^T \quad \text{where} \quad \boldsymbol{Q} = [q_1 \dots q_p] \quad \text{and} \quad \Lambda = \text{diag}(\lambda_1, \dots, \lambda_p) \quad \text{with} \quad \lambda_1 > \dots > \lambda_p > 0$$

As such, if we **want to maximize** the loss function $\ell(\phi_1) = \phi_1^T C \phi_1$ then we should take $\phi_1 = q_1$ so to have $\ell(\phi_1) = \lambda_1$

The principal component is then defined as

$$Z_1 = q_{11}X_1 + \dots + q_{1p}X_p$$

and the **projected data points** over the direction of the principal component are

$$\begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix} q_1 = \boldsymbol{X} q_1 \in \mathbb{R}^{n \times 1}$$

> **Figure 4**: Example on the figure with simulated data

-- **Second-axis of PCA**

We can continue our analysis and search for a **second** principal component, Z_2.

We define it as being the linear combination of the predictors with **maximal variance** while being **uncorrelated** to Z_2

- We can show that asking Z_1 and Z_2 to be **uncorrelated** is the same as asking the principal directions to be **orthogonal**

The projected data points over the second principal direction are then

$$z_{i2} = \sum_{j=1}^{p} \phi_{j2} x_{ij} \quad \text{with} \quad \phi_1^T \phi_2 = 0$$

The optimization problem for ϕ_2 is almost **the same** as for ϕ_1 except for the orthogonality of the vector.

We know that C has an orthonormal eigenvector basis, so we can show that $\phi_2 = q_2$

We can continue this procedure for the p next principal components $Z_3, \dots, Z_p$ with all of them **pairwise uncorrelated**.

- Note that there are **at most** p principal components to consider for a dataset with p predictors

-- **The projected data points**

To **summarize**.

If we want to reduce the dimensionality of a dataset $X \in \mathbb{R}^{n \times p}$ into a new one $Z \in \mathbb{R}^{n \times k}$ where each x_i is related to z_i we do:

- Obtain k principal directions of interest $q_1, \dots, q_k$ from the eigenvectors of $C = \dfrac{1}{n} X^T X$
- Form the projection matrix $Q = [q_1 \dots q_k] \in \mathbb{R}^{p \times k}$
- Obtain the new dataset as

$$Z = XQ \quad \text{with each} \quad z_{ij} = x_i^T q_j$$

-

> **Figure 5**: Show the results of two plots for the `economics` dataset
>
> - This dataset shows some economic indicators for 12 countries

> - It is **hard** to conclude anything about the samples based on so many features
> - Left plot shows the projected data points on the PC1 vs PC2 axis, i.e. the z_i
>   - We see that Brazil and USA are rather **far from the bulk** of other countries
>   - We see that the **European countries** form kind of a cluster
>   - But how can the principal directions **be interpreted** ?
> - Table on the right shows the **projection** matrix with the two first principal components
>   - It is the Q matrix with dimensions $n \times 2$
>   - Note that $z_{i1} = x_i^T q_1 = Q_{11} x_{i1} + Q_{21} x_{i2}$ and $z_{i2} = x_i^T q_2 = Q_{12} x_{i1} + Q_{22} x_{i2}$
>   - The loadings indicate how **each feature contributes** to coordinates of the projected sample
>   - PC1 mainly captures population and area
>   - PC2 mainly captures inflation and gnb.capita
>   - We can interpret things as:
>   - Brazil is a large country (big area and big population) with a very high inflation
>   - USA is also a large country but with controlled inflation
>   - The European countries are small and have a high GNB per capita
>   - USA has a very negative external trade, so is on the opposite direction to EXTTR

-- **How many principal components should we keep?**

We **could stop** our lecture on PCA here and you would already have learned the basics of this very important tool for data analysis.

However, there are still a few things that I would like to present you.

Firstly, when can we say that with have **enough principal components** to represent the data?

Remember that our goal with PCA was to **obtain a new** dataset $Z \in \mathbb{R}^{n \times k}$ that **preserved a maximum** amount of variance from the initial dataset $X \in \mathbb{R}^{n \times p}$

The total initial variance in X can be calculated as (assuming re-centered data)

$$\text{Var}(X) = \sum_{j=1}^{p} \text{Var}(X_j) = \sum_{j=1}^{p} \left(\frac{1}{n} \sum_{i=1}^{n} x_{ij}^2 \right) = \frac{1}{n} \sum_{j=1}^{p} \sum_{i=1}^{n} x_{ij}^2 = \text{tr} \left(\frac{1}{n} X^T X \right) = \frac{1}{n} \sum_{\ell=1}^{p} \lambda_\ell^2$$

and, equivalently, the variance in the Z dataset is

$$\text{Var}(Z) = \text{tr} \left(\frac{1}{n} Z^T Z \right) = \text{tr} \left(\frac{1}{n} Q^T X^T X Q \right) = \frac{1}{n} \sum_{\ell=1}^{k} \lambda_\ell^2$$

So the **proportion of variance** captured by a transformed dataset with k principal components is:

$$\text{PV(k)} = \frac{\sum_{\ell=1}^{k} \lambda_\ell^2}{\sum_{\ell=1}^{p} \lambda_\ell^2}$$

- Note that if $k = p$ then there is no dimensionality reduction and the new dataset Z preserves all of the information from X, i.e. $\text{PV}(k) = 1$
- Note that there at most $\min(n, p)$ principal components when dealing with dataset $X \in \mathbb{R}^{n \times p}$
    - We often think only of cases where $n \gg p$ but you will see a different situation in TP2

-- Principal components regression

Finally, I would like to show you an **important application** of PCA in the context of multiple linear **regression**. You guys will be doing this in your TP2 next week.

Suppose we had a dataset $D = \{(x_i, y_i)\}_{i \leq n}$ with each $x_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$

- We will suppose that both the predictors and the observed variables have been re-centered

A linear regression model for this case would then be

$$y_i = \sum_{j=1}^{p} \beta_j x_{ij} + \varepsilon_i \quad \rightarrow \quad \boldsymbol{y} = \boldsymbol{X}\beta \quad \rightarrow \quad \hat{\beta}_{\text{LS}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

Suppose, however, that p is large and that we know that there are **correlations** between some of the predictors. In this case, we could be interested in obtaining **new predictors** that are linear combinations of the initial predictors.

This is a clear application for PCA !

We can define $k \ll p$ new predictors and write a new regression model as

$$y_i = \sum_{j=1}^{k} \gamma_j z_{ij} + \varepsilon_i \quad \rightarrow \quad \boldsymbol{y} = \boldsymbol{Z}\gamma \quad \rightarrow \quad \hat{\gamma} = (\boldsymbol{Z}^T\boldsymbol{Z})^{-1}\boldsymbol{Z}\boldsymbol{y}$$

And writing the model back into its initial predictors, we have

$$\boldsymbol{Z} = \boldsymbol{X}Q \quad \text{and} \quad \boldsymbol{y} = \boldsymbol{Z}\gamma = \boldsymbol{X}(Q\gamma) \quad \rightarrow \quad \beta = Q\gamma \quad \rightarrow \quad \hat{\beta}_{\text{PCR}} = Q\,\hat{\gamma}$$

where $Q \in \mathbb{R}^{p \times k}$

- Note that when $n < p$, we can not invert X^TX and, therefore, $\hat{\beta}_{\text{LS}}$ is impossible to get
- However, if we use PCA, we can reduce the number of predictors to get down to a situation in which one can invert Z^TZ and obtain the $\hat{\beta}_{\text{PCR}}$

-- **PCA and geometry**

OK, since we still have a few minutes, I would like to show you an **important remark** about PCA.

Suppose that **instead** of searching for a direction in which the projected datapoints had **maximal variance** I had told you that we would search for a direction such that the projected data points were the **closest** to the original ones.

In other words, we want to minimize:

$$\underset{\|\phi\|^2=1}{\text{minimize}} \; \frac{1}{n}\sum_{i=1}^{n} \left\|x_i - (\phi^T x_i)\phi\right\|^2$$

we can rewrite the terms in the sum as

$$\|x_i - (\phi^T x_i)\|^2 = \big(x_i - (\phi^T x_i)\phi\big)^T \big(x_i - (\phi^T x_i)\phi\big) = x_i^T x_i - 2(\phi^T x_i)^2 + (\phi^T\phi)(\phi^T x_i)^2 = \|x_i\|^2 - (\phi^T x_i)^2$$

and rewriting the minimization function we get

$$\underset{\|\phi\|^2=1}{\text{maximize}} \; \frac{1}{n}\sum_{i=1}^{n} (\phi^T x_i)^2$$

which is exactly the maximization of variance that we had asked at the beginning.

This means that in PCA we are at **the same time** looking for a direction in which

- The projected data points have maximum variance along the direction
- The projected data points are the closest to their original counterparts

If we continue this analysis to a PCA with k principal directions, we have that the projected data points over the subspace generated by the vectors $q_1, \dots, q_k$ is written as

$$\tilde{x}_i = \sum_{\ell=1}^{k} (x_i^T q_\ell)\, q_\ell \rightarrow \tilde{X} = (XQ)Q^T$$

-- **SVD and PCA**

It is an important result from linear algebra that a matrix $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ with full column rank can be decomposed as:

$$\boldsymbol{X} = \boldsymbol{U}\, \boldsymbol{\Sigma}\, \boldsymbol{V}^T = [\boldsymbol{u}_1 \dots \, \boldsymbol{u_p}] \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_p \end{bmatrix} [\boldsymbol{v}_1 \dots \, \boldsymbol{v}_p]^T = \sum_{k=1}^{p} \sigma_k\, \boldsymbol{u}_k \boldsymbol{v}_k^T$$

with $\boldsymbol{U} \in \mathbb{R}^{n\times p}$ and $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_r$, $\boldsymbol{V} \in \mathbb{R}^{p\times p}$ and $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}_p$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{p\times p}$ a diagonal matrix with $\boldsymbol{\Sigma}_{ii} = \sigma_i$ the singular values of the matrix.

When calculating the PCA directions, we have to calculate the eigenvectors of

$$X^T X = V \Sigma^2 V$$

therefore the k-principal directions are simply the k vectors of matrix V associated to the largest squared singular values; i.e. $Q = V_k$

Also, the projected data points after a k dimensional PCA are simply

$$\tilde{X}_k = \sum_{\ell=1}^{k} \sigma_\ell\, u_\ell v_\ell^T$$