CHAPTER 8

Principal Components

Principal component analysis (PCA) is one of the most often used tools in statistics (data science, machine learning). It transforms a matrix of observations of possibly correlated variables into a matrix of values of linearly uncorrelated variables called principal components (PC), where each PC is defined by a combination of the columns of the original matrix. The first principal component accounts for as much of the variability in the data as possible, and each subsequent PC has the highest variance possible such that it is orthogonal to all preceding PC. When you are given a matrix large enough not to be able to print on a sheet of A4, the first step in understanding it should involve PCA or its regularised variant.

Example 8.1. Let $A \in \mathbb{R}^{n \times n}$ denote a data matrix, which encodes $n$ ratings of $n$ users, e.g., of movies or books. Let us assume the ratings of each user, i.e., values of each column, sum up to 1. The first 2 principal components could represent a new coordinate system with two axes, such as “likes horrors” and “likes romantic comedies”, depending on the actual ratings. You could replace the ratings of one user in the original rating-per-movie form, i.e., one row, with a 2-vector, which would suggest how much the user likes horrors and how much the user likes romantic comedies. For excellent interactive demonstrations, see: http://setosa.io/ev/principal-component-analysis/

You should remember: Principal Component Analysis is a linear transformation that transforms data into a new coordinate system such that the projection of the data on the first coordinate explains more variance in the data than the projection on the the second coordinate, which in turn explains more variance than a projection onto the third coordinate etc. The Eigenvalue Problem: given $A \in \mathbb{R}^{n \times n}$, find the unknowns $x \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}$ such that $Ax = \lambda x$. Here, $\lambda$ is called an eigenvalue of $A$ and $x$ is an eigenvector of $A$. Power Method for computing the greatest eigenvalue by absolute value, based on $x^{k+1} = \frac{Ax^k}{\|Ax^k\|}$.

1. Principal Component Analysis

Let $A \in \mathbb{R}^{n \times n}$ denote a data matrix, which encodes $m$ observations (samples) of dimension $n$ each (e.g., $n$ features), which has been normalised such that each column has zero mean. PCA extracts convex combinations of columns of $A$ while maximising $\ell_2$ norm of $Ax$. The extraction of the first combination is:

$$\max_{x \in \mathbb{R}^n} \|Ax\|_2 \text{ such that } \|x\|_2 \leq 1,$$

(8.1)

The optimum $x^* \in \mathbb{R}^n$ is called the loading vector. $Ax^*$ is called the first principal component. Each row-vector $A_i$ of $A$ is mapped to a new vector $t = x \cdot A_i$ in terms of the principal component $x$. One can consider further principle components, usually sorted in the decreasing order of the amount of variance they explain. These can be obtained by running the same methods on an updated matrix $A^{k+1} = A^k - x^k(x^k)^T A^k x^k (x^k)^T$, which is known as Hotelling’s deflation. The combinations point in mutually orthogonal directions.

Theorem 8.2 (Hotelling). If $\lambda_1 \geq \ldots \geq \lambda_p$ are the eigenvalues of symmetric positive semidefinite matrices $A \in \mathbb{R}^{n \times n}$, $x_1, \ldots, x_n$ are the corresponding eigenvectors, and $\bar{A} = A - x_j x_j^T A x_j x_j^T$ for some $1 \leq j \leq n$, then $\bar{A}$ has eigenvectors $x_1, \ldots, x_n$ with corresponding eigenvalues $\lambda_1, \ldots, \lambda_{j-1}, 0, \lambda_{j+1}, \lambda_n$.

For a proof, see Mackey [2009].
For a symmetric matrix, e.g., $X^TX$, $x^*$ is eigenvector corresponding to the largest eigenvalue. The largest eigenvalue is equivalent to the objective function value at $x^*$. As a proportion of the sum of the eigenvalues, it is also a measure of the variance explained by the first eigenvalue. Let us prove this:

**Proof sketch.** Consider:

$$\max_{x \in \mathbb{R}^n} \|Ax\|_2 \text{ such that } \|x\|_2 \leq 1,$$

where $\|\cdot\|_2$ is $\ell_2$. The Lagrangian is $L(x) = x^TAx - \lambda (x^Tx - 1)$ where $\lambda \in \mathbb{R}$ is a newly introduced Lagrangian multiplier. The stationary points of $L(x)$ satisfy

$$\frac{dL(x)}{dx} = 0 \quad (8.3)$$

$$2x^TA^T - 2\lambda x^T = 0 \quad (8.4)$$

$$Ax = \lambda x. \quad (8.5)$$

Recalling $Ax = \lambda x$ in the definition of an eigenvalue problem, each eigenvalues $\lambda$ of $A$ is hence the value of $L(x)$ at an eigenvectors of $A$.

### 2. Eigenvalues and Eigenvectors

Notice that we have seen eigenvalues a number of times before: We have seen that positive definite matrices $A$, i.e., $x^TAx > 0$ for all $x \neq 0$, have $x^TAx = \lambda x^Tx > 0$ and eigenvalues $\lambda > 0$. We have seen the set $\sigma(A)$ of all eigenvalues of $A$, called the spectrum. The absolute value of the dominant eigenvalue is the spectral radius, which we have seen in the definition of a contraction mapping. We have also seen the spectral condition number of symmetric $A$, $\text{cond}_{\text{rel}}(A) := \frac{\max_{\lambda \in \sigma(A)} |\lambda|}{\min_{\lambda \in \sigma(A)} |\lambda|}$, which was a measure of the distortion produced by $A$, i.e., the difference in expansion/contraction of eigenvectors that $A$ can cause.

Prior to explaining more about eigenvalues and eigenvectors, let us see an interactive demonstration: [http://setosa.io/ev/eigenvectors-and-eigenvalues/](http://setosa.io/ev/eigenvectors-and-eigenvalues/) and a piece of Python libraries for computing eigenvalues and eigenvectors:

```python
import numpy as np
A = np.random.rand(2,2)
(vals, vecs) = np.linalg.eig(A)
np.dot(A, vecs[:,0]), vals[0] * vecs[:,0]
np.dot(A, vecs[:,1]), vals[1] * vecs[:,1]
```

Let us now present some more properties of eigenvalues: The $i$th eigenvalue functional $A \mapsto \lambda_i(A)$ is not a linear functional beyond dimension one. It is not convex (except for $i = 1$) or concave (except for $i = n$). For any $m \geq 5$, there is an $m \times m$ matrix with rational coefficients whose eigenvalue cannot be written using any expression involving rational numbers, addition, subtraction, multiplication, division, and taking $k$th roots. The proof is based on the following:

**Example 8.3 (Trefethen and Bau [1997]).** Consider the polynomial $p(z) = z^m a_{m-1} z^{m-1} + \cdots + a_1 z + a_0$. The roots of $p(z)$ are equal to the eigenvalues of:

$$\begin{bmatrix}
0 & -a_0 \\
1 & -a_1 \\
\vdots & \vdots \\
1 & -a_{m-2} \\
\vdots & \vdots \\
1 & -a_{m-1}
\end{bmatrix}$$

and
THEOREM 8.4 (Abel). For any $m \geq 5$, there is a polynomial $p(z)$ of degree $m$, with rational coefficients that has a real root $p(r)$ with the property that $r$ cannot be written using any expression involving rational numbers, addition, subtraction, multiplication, division, and taking $k$th roots.

On the other hand: The $i$th eigenvalue functional $A \mapsto \lambda_i(A)$ is Lipschitz continuous for symmetric matrices, for fixed $1 \leq i \leq n$. There is a characterisation, resembling convexity:

THEOREM 8.5 (Courant-Fischer). Let $A \in \mathbb{R}^{n \times n}$ be symmetric. Then we have

$$\lambda_i(A) = \sup_{\dim(V) = i} \inf_{v \in V: \|v\| = 1} v^T Av$$

and

$$\lambda_i(A) = \inf_{\dim(V) = n-i+1} \sup_{v \in V: \|v\| = 1} v^T Av$$

for all $1 \leq i \leq n$, where $V$ ranges over all subspaces of $\mathbb{R}^n$ with the indicated dimension.

Finally, there are a variety of good news in perturbation analysis. Let us consider a symmetric matrices $A, B \in \mathbb{R}^{n \times n}$ and view $B$ as a perturbation of $A$. One can prove the so called Weyl inequalities:

$$\lambda_{i+j-1}(A + B) \leq \lambda_i(A) + \lambda_j(B), \quad (8.6)$$

for all $i, j \geq 1$ and $i + j - 1 \leq n$, the so called Ky Fan inequality

$$\lambda_1(A + B) + \ldots + \lambda_k(A + B) \leq \lambda_1(A) + \ldots + \lambda_k(A) + \lambda_1(B) + \ldots + \lambda_k(B) \quad (8.7)$$

and the Tao inequality:

$$|\lambda_i(A + B) - \lambda_i(A)| \leq \|B\|_{op} = \max(|\lambda_1(B)|, |\lambda_n(B)|). \quad (8.8)$$

These suggest that for symmetric matrices, the spectrum of $A + B$ is close to that of $A$ if $B$ is small in operator norm.

EXAMPLE 8.6. Google started out with a patent application for PageRank. There $A \in \mathbb{R}^{n \times n}$ is an “adjacency” matrix, which links between pairs of websites. $n$ is the number of websites. The eigenvector corresponding to the dominant eigenvalue suggested a reasonable rating of the websites for the use in search results. There, however, you need a very efficient method of computing the eigenvector, considering $n$ is very large. \hfill \diamondsuit

3. Power Method

The power method is a simple iterative algorithm for computing the largest eigenvalue by absolute value (8.2). It is based on $x^{k+1} = \frac{Ax^k}{\|Ax^k\|}$, which is motivated by the fact that given a matrix $A$ and a basis of $\mathbb{R}^n$ comprising eigenvectors of $A$, vectors multiplied by $A$ are expanded most in the direction of the eigenvector associated with the eigenvalue of the biggest magnitude.

EXAMPLE 8.7. Let $A$ be a $2 \times 2$ matrix with eigenvalues $3$ and $1/2$ and corresponding eigenvectors $y$ and $z$. Recall that eigenvectors of distinct eigenvalues are always linearly independent. We can hence think of $\{y, z\}$ as a basis of $\mathbb{R}^2$. Let $x^0$ be any linear combination of $y$ and $z$, e.g., $x^0 = y + z$. (In general, we could study $x^0 = \alpha y + \beta z$ for arbitrary scalars $\alpha, \beta$ but here we are just letting $\alpha$ and $\beta$ both equal $1$, to simplify the discussion.)

Now let $x^1 = Ax^0$, and $x^{k+1} = Ax^k = A^k x_0$ for each $k \in \mathbb{N}$.

Since matrix multiplication is distributive over addition,

$$x^1 = Ay + Az = 3y + \frac{1}{2}z.$$ 

Thus the $y$ component of $x^0$ is expanded while the $z$ component is contracted. Repeating this process $k$ times, we get:

$$x^k = Ax^{k-1} = A^k x^0 = 3^k y + \frac{1}{2^k} z.$$ 

Thus, $x^k$ expands in the $y$ direction and contracts in the $z$ direction. Eventually, $x^k$ will be almost completely made up of its $y$ component, and the $z$ component will be negligible.
Note that how many iterations $k$ are needed depends on how much the biggest eigenvalue exceeds the second biggest in magnitude. In the example, the ratio is $\frac{3}{\sqrt{2}} = 6$ so expansion in the $y$ direction is happening 6 times faster than in the $z$ direction.

In general, for any matrix $A$, the growth rate of its powers $A^k$ as $k \to \infty$ depends on the eigenvalue(s) of $A$ whose absolute value is greatest: this largest absolute value is of course the spectral radius $\rho(A)$. These eigenvalue(s) are generally called the dominant eigenvalue(s) of $A$.

Let us generalise the example. Denote the approximation to the eigenvector at iteration $k$ by $x^k$. The initial $x^1$ may be chosen randomly or set to an approximation to a dominant eigenvector, if there is some knowledge of this. The core of the method is the step

$$x^{k+1} := \frac{1}{\|Ax^k\|}Ax^k.$$ 

That is, at each iteration, $x^k$ is left-multiplied by $A$ and normalised (divided by its own norm) giving a unit vector in the direction of $Ax^k$. If $x^k$ were an eigenvector of $A$, then $x^{k+1}$ would be equal to $x^k$ (since they are both unit vectors, both in the same direction). This suggest a Cauchy-type convergence criterion.

Choose the starting $x^1$ so that $\|x^1\| = 1$.

```python
import numpy as np

def Power(A, x0, tol = 1e-5, limit = 100):
    x = x0
    for iteration in xrange(limit):
        next = A*x
        next = next/np.linalg.norm(next)
        if allclose(x, next, atol=tol):
            break
        x = next
    return next
```

The method will converge if:

- $A$ has a dominant real eigenvalue $\lambda_1$, that is, $|\lambda_1|$ is strictly larger than $|\lambda|$ for any other eigenvalue $\lambda$ of $A$,
- the initial vector has a nonzero component in the direction of the eigenvector $x$ corresponding to $\lambda_1$.

Certain classes of matrices are guaranteed to have real eigenvalues, e.g., symmetric matrices, positive definite matrices. Also, the adjacency matrix of a connected graph will have a dominant real eigenvalue, by the Perron-Frobenius theorem. On the other hand, the power iteration method will fail if $A$’s dominant eigenvalue(s) have non-zero imaginary parts.

**Theorem 8.8** (Trefethen and Bau [1997]). Assuming $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_m| \geq 0$ and non-zero $x^0$, the iterates of power method satisfy:

$$\|x^k - (\pm x^*)\| = O\left(\frac{|\lambda_2|}{\lambda_1}^k\right)$$  \hspace{1cm} (8.9)

$$|(x^k)^T Ax^k - \lambda_1| = O\left(\frac{|\lambda_2|}{\lambda_1}^{2k}\right)$$  \hspace{1cm} (8.10)

as $k \to \infty$. The $\pm$ indicates that at each iteration $k$, the bound holds for one of the signs.
So its convergence is very fast, except when the largest and second largest eigenvalues of $A$ are very close in absolute value.

Assuming $A$ is dense, the worst case complexity per iteration is $n^2$ multiplications in the computation of $Ax^k$. Computing the vector norm $\|Ax^k\| = \sqrt{(Ax^k) \cdot (Ax^k)}$ also takes $n$ multiplications. This gives a total of $O(n^2)$ operations. Assuming $A$ is sparse, such as in PageRank computations, the number of floating point operations in the computation of $Ax^k$ is twice the number of non-zeros in $A$, independent of $n, m$.

Further, one may note that:

- Once the dominant eigenvalue is found, the deflation method suggested by Theorem 8.2 can be used to find the second largest eigenvalue, etc. For finding all eigenvalues, decomposition methods may be preferrable, though.
- If the initial $x^1$ had a zero component in the $x$ direction, we are reliant on rounding errors to get a component in the $x$ direction at some stage in the run. Once we get a non-zero $x$ component, it will expand relative to the other components, but we may need to wait a while for this component to appear and then grow.
- The expression $Ax$ could be replaced based on $Ax = \lambda x \Rightarrow x^t Ax = \lambda x^t x = \lambda \|x\|^2 \Rightarrow \lambda = \frac{1}{\|x\|^2} x^t Ax$.
  which makes it possible to introduce additional checks to avoid under- and over-flows and division by very small numbers.
- Otherwise, one only needs to store $A$ and two vectors $x^k$ and $Ax^k$. This is possible even for $n \times n$ matrix $A$, $n$ in the billions, such as for the Google PageRank.