Lec 6: Eigenvalue Decomposition

Ailin Zhang

2022-09-28

Agenda

- Eigenvalue decomposition
- Power Method
- Matrix form of power method
- Principal component analysis

Eigenvalue decomposition and diagonalization

A $p \times p$ symmetric matrix Σ can be diagonalized by $\Sigma = Q\Lambda Q^\top$

where Q is an orthogonal matrix, and Λ is a diagonal matrix, $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, ..., \lambda_p)$, where we order λ_j from largest to smallest in magnitude for $j = 1, ..., p$.

$$\Sigma = Q\Lambda Q^\top = Q \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_p \end{bmatrix} Q^\top$$

$\Sigma Q = Q\Lambda$, so $\Sigma q_j = \lambda_j q_j$. The column vectors in Q are eigenvectors. The diagonal elements in Λ are eigenvalues of Σ.

Power Method (Some preparation)

For a vector $\vec{v}$, let $\vec{u}$ be its **coordinates** in system Q, i.e. $\vec{v} = Q\vec{u}$

$$\vec{v} = Q\vec{u} = \left[Q_1, Q_2, \cdots, Q_n\right] \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} = u_1 Q_1 + u_2 Q_2 + \cdots + u_n Q_n$$

if we left multiply $Q^\top$ on the two sides of the equation: $\vec{u} = Q^\top \vec{v}$

$u_i = < \vec{v}, Q_i >$

Power Method

$v = Qu \quad \Sigma = Q\Lambda Q^\top$

$$\Sigma v = Q\Lambda Q^\top Qu = Q \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_p \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_p \end{bmatrix} = Q \begin{bmatrix} \lambda_1 u_1 \\ \lambda_2 u_2 \\ \vdots \\ \lambda_p u_p \end{bmatrix},$$

which means the vector Σv becomes $(\lambda_1 u_1, \lambda_2 u_2, \cdots, \lambda_p u_p)^\top$ in basis Q, i.e., Σ is Λ in Q.

If we repeat this process n times, then Σ^n is Λ^n in Q,

$$v \xrightarrow{\Sigma^n} (\lambda_1^n u_1, \lambda_2^n u_2, \cdots, \lambda_p^n u_p)^\top.$$

We can keep normalizing $v \leftarrow v/|v|$ to make v a unit vector in this process.

Question: what happens if $n \to \infty$?

Power Method

Suppose λ_1 has the greatest magnitude, this procedure will converge to $u = (1, 0, \cdots, 0)$ in the space of Q, and the corresponding $v = q_1$.

The power method iterates the following two steps:

- Compute normalized vector $\tilde{v} = \dfrac{v}{|v|}$.
- Update $v = \Sigma \tilde{v}$.

Question: How to get q_2?

To get q_2 using this method, we initialize the above procedure with a vector $v \perp q_1$ that is perpendicular to q_1. In Q, the first component of u will always be 0, then the procedure will converge to $u = (0, 1, 0, \cdots, 0)$ in the space of Q, and the corresponding $v = q_2$.

Question: How to get q_3?

Power Method

To get q_3, we initialize the above procedure with v perpendicular to both q_1 and q_2, i.e. $v \perp q_1$ and $v \perp q_2$.

Continue the above procedure, we eventually get all the vectors in Q.

Matrix form of power method

We can parallelize the above sequential method, by starting from p vectors $V = (V_1, ..., V_p)$ and maintain their orthogonality after each multiplication by Σ, by iterating the following two step

- Compute $\tilde{V}$, the orthogonalized V.
- Update $V = \Sigma \tilde{V}$.

Python Code

```
def eigen_qr(A):
    T = 1000
    A_copy = A.copy()
    r, c = A_copy.shape

    V = np.random.random_sample((r, r))

    for i in range(T):
        Q, _ = qr(V)
        V = np.dot(A_copy, Q)
    Q, R = qr(V)
    return R.diagonal(), Q
```

Principle Component Analysis (PCA)

Consider the $n \times p$ data matrix $\mathbf{X}$. Let us assume that all the columns of $\mathbf{X}$ are centralized, i.e., $\sum_{i=1}^{n} x_{ij}/n = 0$.

In other words, let $\mathbf{1}$ be the $n \times 1$ column vector of 1's. Then $\langle \mathbf{X}_j, \mathbf{1} \rangle = 0$, for $j = 1, ..., p$, i.e., $\mathbf{1}^\top \mathbf{X} = 0$.

For each row of $\mathbf{X} = (X_1^\top, ..., X_n^\top)^\top$, we want to represent observation X_i in a new basis system Q, so that $X_i = QZ_i$.

Let $\mathbf{Z} = (Z_1^\top, ..., Z_n^\top)^\top$ be the data matrix in Q. We want the columns of $\mathbf{Z} = (\mathbf{Z}_1, ..., \mathbf{Z}_p)$ to be orthogonal to each other, so that they are uncorrelated.

If you regress any column of $\mathbf{Z}$ on another column of $\mathbf{Z}$, the regression coefficient is 0. Let $\lambda_j = \|\mathbf{Z}_j\|^2/n = \sum_{i=1}^{n} z_{ij}^2/n$, then λ_j is the variance of $\{z_{ij}, i = 1, ..., n\}$, and $\mathbf{Z}^\top \mathbf{Z} = \Lambda = \mathrm{diag}(\lambda_1, ..., \lambda_p)$. Then

$$\mathbf{X}^\top \mathbf{X} = Q\mathbf{Z}^\top \mathbf{Z} Q^\top = Q\Lambda Q^\top.$$

Principle Component Analysis (PCA)

$$\mathbf{X}^\top \mathbf{X} = Q\mathbf{Z}^\top \mathbf{Z} Q^\top = Q\Lambda Q^\top.$$

Question: How to solve Q and Λ?

We can use the power method to compute Q and Λ.

We can choose $d < p$, and represent $X_i \approx \sum_{k=1}^{d} z_{ik} q_k$. This is **principal component analysis for dimension reduction**.

Python Code

```
n = 100
p = 5
X = np.random.random_sample((n, p))
A = np.dot(X.T, X)

D, V = eigen_qr(A)
print (D.round(6))
print (V.round(6))

# Compare the result with the numpy calculation
eigen_value_gt, eigen_vector_gt = np.linalg.eig(A)
print (eigen_value_gt.round(6))
print (eigen_vector_gt.round(6))
```

Singular Value Decompositon (SVD)

It generalizes the eigenvalue decomposition of a square normal matrix with an orthonormal eigenbasis to any $m \times n$ matrix.

$$M = U\Sigma V^\top$$

$$M = \begin{bmatrix} u_1 & u_2 & \cdots & u_m \end{bmatrix} \begin{bmatrix} \lambda_1 & & & & & \\ & \lambda_2 & & & & \\ & & \ddots & & & \\ & & & \lambda_p & & \\ & & & & \ddots & \\ & & & & & 0 \end{bmatrix} \begin{bmatrix} v_1^\top \\ v_2^\top \\ \vdots \\ v_n^\top \end{bmatrix}$$

Question for you: How do you solve for U, Σ, *and* V?