Power iteration, inverse iteration, and Rayleigh quotient iteration

Jing Li, Kent State University

1 Rayleigh quotient

Let us first consider a linear least squares problem. We know in general, given a vector \( b \in \mathbb{R}^m \), and a matrix \( A \in \mathbb{R}^{m \times n} \), where \( rank(A) = n \), the best fit to \( b \) from the range of \( A \) is \( Ax \), where \( x \) is the linear least square solution and can be found through solving the normal equation

\[
ATAx = ATb.
\]

Here we consider such a linear least squares problem, but in a different setting. Let \( A \) be a real symmetric \( n \times n \) matrix. Given any vector \( x \in \mathbb{R}^n \), in general \( x \) is not necessarily an eigenvector of \( A \), i.e., \( Ax \) is not a multiple of \( x \). But we can consider the best fit to \( Ax \) among all multiples of \( x \), i.e., we best fit \( Ax \) from the range of \( x \) and solve the following linear least squares problem

\[
\alpha^* = \arg \min_{\alpha \in \mathbb{R}} \|x\alpha - Ax\|_2.
\]

In such case, \( x \in \mathbb{R}^{n \times 1} \) is the coefficient matrix in the linear least squares problem, \( Ax \) is the given right hand side vector. The solution \( \alpha^* \) satisfies the following normal equation

\[
x^T\alpha^* = x^TAx,
\]

i.e.,

\[
\alpha^* = \frac{x^TAx}{x^Tx}.
\]

Then \( \alpha^*x \) is closest to the vector \( Ax \) among all possible scalar multiples of \( x \). If \( x \) is indeed an eigenvector of \( A \) corresponding to an eigenvalue \( \lambda \), i.e., \( Ax = \lambda x \). Then we can see that

\[
\alpha^* = \frac{x^TAx}{x^Tx} = \frac{x^T\lambda x}{x^Tx} = \lambda.
\]

If \( x \) is not an eigenvector of \( A \), but is close to an eigenvector of \( A \), then \( \alpha^* \) can still be used as an approximation to an eigenvalue of \( A \).
In particular, given a real symmetric matrix $A \in \mathbb{R}^{n \times n}$, we call
\[
    r(x) = \frac{x^T A x}{x^T x},
\]
the Rayleigh quotient of $A$ at any vector $x \in \mathbb{R}^n$. Here we treat the Rayleigh quotient $r(x)$ as a function of $x$, corresponding to a given matrix $A$. If $x$ is an eigenvector of $A$, then its Rayleigh quotient $r(x)$ is just the corresponding eigenvalue. If $x$ is close to an eigenvector $q$ of the matrix $A$, corresponding to an eigenvalue $\lambda$, then its Rayleigh quotient $r(x)$ should be a good approximation of $\lambda$, which can in fact be analyzed in the following.

From the Taylor expansion of function of multi variables, we know that the Taylor expansion of $r(x)$ at $q$ is
\[
    r(x) = r(q) + \nabla r(q)^T (x - q) + \frac{(x - q)^T \nabla^2 r(\bar{q})(x - q)}{2}.
\]
Here $q$ is an eigenvector of $A$ corresponding to an eigenvalue $\lambda$. $\nabla r(q)$ is the Jacobian of the function $r(x)$ at $x = q$, and $\nabla^2 r(\bar{q})$ represents the Hessian matrix of the function $r(x)$ at $\bar{q}$ which is close to $q$. The Jacobian $\nabla r(x)$ is a column vector in this case and contains the partial derivative of $r(x)$ with respect to the components of $x$. From the definition $r(x) = \frac{x^T A x}{x^T x}$, we have
\[
    \frac{\partial r(x)}{\partial x_k} = \frac{\partial (x^T A x)}{\partial x_k} = \frac{2(A x)_k (x^T x) - x^T A (2 r)_k}{(x^T x)^2} = \frac{2}{x^T x} (A x - r(x)) x_k.
\]
Then we have
\[
    \nabla r(x) = \frac{2}{x^T x} (A x - r(x)) x.
\]
At the eigenvector $q$, we know $r(q) = \lambda$, and we have
\[
    \nabla r(q) = \frac{2}{q^T q} (A q - \lambda q) = 0.
\]
Then the Taylor expansion of $r(x)$ at $q$ becomes
\[
    r(x) = \lambda + \frac{(x - q)^T \nabla^2 r(\bar{q})(x - q)}{2}.
\]
Assuming that $\|\nabla^2 r(\bar{q})\|$ is bounded by a constant, we have
\[
    |r(x) - \lambda| = O (\|x - q\|^2).
\]
This tells us that for a given symmetric matrix $A$, if $x$ is an approximation of an eigenvector of $A$, then its Rayleigh quotient $r(x)$ approximates the corresponding eigenvalue of $A$. The accuracy is even better than the approximation of the eigenvector.

If $x$ is normalized, i.e., if $\|x\|_2 = 1$, then the Rayleigh quotient of $x$ is $r(x) = x^TAx$.

2 A few properties of eigenvalues

- **Inversion** If $A$ is nonsingular and $Ax = \lambda x$, for $x \neq 0$, then $A^{-1}x = \frac{1}{\lambda}x$. In another word, if $\lambda$ is an eigenvalue of $A$, then $\lambda^{-1}$ is an eigenvalue of $A^{-1}$. Their eigenvectors are the same.

- **Powers** If $Ax = \lambda x$, then for any integer $k$, $A^kx = \lambda^kx$, i.e., taking the $k$th power of a matrix also takes the $k$th power of its eigenvalues, while the eigenvectors are the same.

- **Shift** If $Ax = \lambda x$, then for any value $\sigma$, $(A - \sigma I)x = (\lambda - \sigma)x$. Thus, the eigenvalues of the matrix $A - \sigma I$ are shifted from the eigenvalues of $A$ by $\sigma$, and again their eigenvectors are unchanged.

3 Power iteration

We know that any real $n \times n$ symmetric matrix $A$, have $n$ real eigenvalues $\lambda_1$, $\lambda_2$, ..., $\lambda_n$, and a set of orthonormal eigenvectors $q_1$, $q_2$, ..., $q_n$, which satisfy

$$Aq_i = \lambda_i q_i, \quad \text{for} \quad i = 1, 2, ..., n.$$  

By default, the eigenvalues follow the order $|\lambda_1| \geq |\lambda_2| \geq ... \geq |\lambda_n|$.

Since $q_1, q_2, ..., q_n$ span the $n$-dimensional vector space, any arbitrary vector $v^{(0)}$ can be written as a combination of $q_1, q_2, ..., q_n$,

$$v^{(0)} = a_1 q_1 + a_2 q_2 + ... + a_n q_n.$$

Then the sequence of vectors, $Av^{(0)}$, $A^2v^{(0)}$, $A^3v^{(0)}$, ..., can be written as

$$A^{k}v^{(0)} = A(a_1 \lambda_1^{k-1} q_1 + a_2 \lambda_2^{k-1} q_2 + ... + a_n \lambda_n^{k-1} q_n) = a_1 \lambda_1^k q_1 + a_2 \lambda_2^k q_2 + ... + a_n \lambda_n^k q_n.$$  

3
\[ a_1 \lambda_1^k \left( q_1 + \frac{a_2}{a_1} \left( \frac{\lambda_2}{\lambda_1} \right)^k q_2 + \ldots + \frac{a_n}{a_1} \left( \frac{\lambda_n}{\lambda_1} \right)^k q_n \right) \]

We can see that if \( |\lambda_1| > |\lambda_2| \geq \ldots \geq |\lambda_n| \), then as \( k \to \infty \), \( \frac{A^k v^{(0)}}{a_1 \lambda_1^k} \to q_1 \) and the speed of convergence is determined by

\[ \left\| \frac{A^k v^{(0)}}{a_1 \lambda_1^k} - q_1 \right\| = O \left( \left| \frac{\lambda_2}{\lambda_1} \right|^k \right) \]

The convergence will be faster if \( |\lambda_2| \) is much smaller than \( |\lambda_1| \), slower if \( |\lambda_2| \) is closer to \( |\lambda_1| \). The convergence rate is called linear, since

\[ \left\| \frac{A^k v^{(0)}}{a_1 \lambda_1^k} - q_1 \right\| = \left\| \frac{A^{k-1} v^{(0)}}{a_1 \lambda_1^{k-1}} - q_1 \right\| O \left( \left| \frac{\lambda_2}{\lambda_1} \right| \right) \]

i.e., \( \frac{A^k v^{(0)}}{a_1 \lambda_1^k} - q_1 \) is reduced by a factor of \( \frac{\lambda_2}{\lambda_1} \) after each step.

From this observation, we can define the following sequence, for given initial guess \( v^{(0)} \), with \( \|v^{(0)}\| = 1 \),

\[ v^{(1)} = \frac{Av^{(0)}}{\|Av^{(0)}\|}, \quad v^{(2)} = \frac{Av^{(1)}}{\|Av^{(1)}\|}, \quad \ldots, \quad v^{(k)} = \frac{Av^{(k-1)}}{\|Av^{(k-1)}\|}, \quad \ldots \]

to approximate the eigenvector corresponding the largest eigenvalue of \( A \).

**Algorithm: Power iteration**

---

Given initial \( v^{(0)} \), with \( \|v^{(0)}\| = 1 \)

for \( k = 1, 2, 3, \ldots \)

\[ v = Av^{(k-1)} \]

\[ v^{(k)} = v / \|v\| \]

\[ \lambda^{(k)} = v^{(k)T} Av^{(k)} \]

end

---

To determine whether the convergence is achieved or not, we can check whether the value \( \|Av^{(k)} - \lambda^{(k)} v^{(k)}\| \) is sufficiently small.
In the power iteration algorithm, $v^{(k)}$ converges to $q_1$, as $k \to \infty$, at the rate
\[
\|v^{(k)} - q_1\| = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right), \quad \text{i.e.,} \quad \|v^{(k)} - q_1\| = \|v^{(k-1)} - q_1\| O\left(\left|\frac{\lambda_2}{\lambda_1}\right|\right),
\]
which is linear. $\lambda^{(k)}$ is the Rayleigh quotient of $v^{(k)}$ and it converges to the corresponding eigenvalue $\lambda_1$ at the rate
\[
|\lambda^{(k)} - \lambda_1| = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right), \quad \text{i.e.,} \quad |\lambda^{(k)} - \lambda_1| = |\lambda^{(k-1)} - \lambda_1| O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^2\right),
\]
which is also linear.

The conditions for the convergence is that $|\lambda_1| > |\lambda_2|$, and the initial vector $v^{(0)}$ has a nonzero component in the direction of eigenvector $q_1$. The second condition should usually be satisfied due to the rounding error in the computation.

4 Inverse iteration

The power iteration only converges to the eigenvector corresponding to the largest eigenvalue of a matrix $A$. The inverse iteration can be used to approximate an eigenvector corresponding to the eigenvalue closest to an initially specified value.

Given any real value $\mu$, the eigenvalues of the matrix $(A - \mu I)^{-1}$ are simply $(\lambda_i - \mu)^{-1}$, $i = 1, 2, ..., n$, where $\lambda_i$ are the eigenvalues of $A$. Let us assume that $\lambda_J$ is closest to $\mu$, and the next closest one is $\lambda_l$. Then we know that the largest eigenvalue of $(A - \mu I)^{-1}$ in absolute value is $(\lambda_J - \mu)^{-1}$, and the second largest one is $(\lambda_l - \mu)^{-1}$, i.e.,
\[
|\lambda_J - \mu|^{-1} > |(\lambda_l - \mu)|^{-1} \geq \cdots
\]
Therefore, we can simply apply the power iteration to the matrix $(A - \mu I)^{-1}$ and it will converge to the eigenvector of the matrix $(A - \mu I)^{-1}$ corresponding to its largest eigenvalue $(\lambda_J - \mu)^{-1}$, which is the same eigenvector of $A$ corresponding to the eigenvalue of $\lambda_J$.

The following inverse iteration algorithm is essentially the power iteration, but applied on the matrix $(A - \mu I)^{-1}$.
Algorithm: Inverse iteration

Given \(v^{(0)}\), with \(\|v^{(0)}\| = 1\), and \(\mu\) (an estimate of eigenvalue \(\lambda_J\)) for \(k = 1, 2, 3, \ldots\)

\[
v = (A - \mu I)^{-1}v^{(k-1)}, \text{ (i.e., solve } (A - \mu I)v = v^{(k-1)} \text{ for } v)\]

\[
v^{(k)} = v/\|v\|\]

\[
\lambda^{(k)} = v^{(k)T}Av^{(k)}
\]

end

Then following the same convergence theory as for the power iteration, we can see that \(v^{(k)}\) converges to \(q_J\), as \(k \to \infty\), at the rate

\[
\|v^{(k)} - q_J\| = O \left( \left| \frac{\lambda_J - \mu}{\lambda_l - \mu} \right|^k \right), \text{ i.e., } \|v^{(k)} - q_J\| = \|v^{(k-1)} - q_J\| O \left( \left| \frac{\lambda_J - \mu}{\lambda_l - \mu} \right|^k \right),
\]

\(\lambda^{(k)}\) converges to \(\lambda_J\) at the rate

\[
|\lambda^{(k)} - \lambda_J| = O \left( \left| \frac{\lambda_J - \mu}{\lambda_l - \mu} \right|^{2k} \right), \text{ i.e., } |\lambda^{(k)} - \lambda_J| = |\lambda^{(k-1)} - \lambda_J| O \left( \left| \frac{\lambda_J - \mu}{\lambda_l - \mu} \right|^{2k} \right).
\]

The convergence rate is also linear and the convergence will be faster if \(\mu\) is closer to \(\lambda_J\).

The inverse iteration is more expensive than power iteration, because it requires the solution of a linear system in each iteration step. Note that the LU or Cholesky factorization of \(A - \mu I\) only needs to be computed once in the algorithm; in each step, only the forward and backward substitutions need to be implemented to solve the system.

5 Rayleigh quotient iteration

In the inverse iteration, \(\lambda^{(k)}\) converges to \(\lambda_J\) and the convergence rate depends on \(\mu\). The convergence will become faster if \(\mu\) is closer to \(\lambda_J\).

In fact \(\lambda^{(k)}\) in the inverse iteration indeed provides a better estimate of \(\lambda_J\) than the initial \(\mu\). If, at each step of the inverse iteration, \(\mu\) is replaced by the latest \(\lambda^{(k)}\), then we have the following Rayleigh quotient iteration.
Algorithm: Rayleigh quotient iteration

Given \( v^{(0)} \), with \( \|v^{(0)}\| = 1 \)
\( \lambda^{(0)} = (v^{(0)})^T A v^{(0)} \)

for \( k = 1, 2, 3, \ldots \)
\[ v = (A - \lambda^{(k-1)} I)^{-1} v^{(k-1)} , \quad \text{(i.e., solve} \ (A - \lambda^{(k-1)} I) v = v^{(k-1)} \text{ for} \ v) \]
\[ v^{(k)} = v / \|v\| \]
\[ \lambda^{(k)} = v^{(k)^T} A v^{(k)} \]
end

The convergence rate of the Rayleigh quotient iteration can be analyzed as the following. At step \( k \) of the iteration, \( v^{(k)} \) is generated essentially by an inverse iteration with the shift value \( \mu = \lambda^{(k-1)} \). Then following the convergence rate of the inverse iteration, we have
\[
\|v^{(k)} - q_J\| = \|v^{(k-1)} - q_J\| O \left( \frac{|\lambda_J - \lambda^{(k-1)}|}{|\lambda - \lambda^{(k-1)}|} \right)
\]
\[
= \|v^{(k-1)} - q_J\| O \left( |\lambda_J - \lambda^{(k-1)}| \right) = O \left( \|v^{(k-1)} - q_J\|^3 \right).
\]
\[
|\lambda^{(k)} - \lambda_J| = O \left( \|v^{(k)} - q_J\|^2 \right) = O \left( \|v^{(k-1)} - q_J\|^6 \right) = O \left( |\lambda^{(k-1)} - \lambda_J|^3 \right).
\]

Therefore the convergence rate of the Rayleigh quotient iteration is at least cubic, since the error at the new step is the cubic of the error at the previous step times a scalar.

Rayleigh quotient converges to a pair of eigenvalue/eigenvector for almost all starting vector \( v^{(0)} \). Its convergence is much faster than the inverse iteration. But on the other hand, it is also more expansive, because at each step, one LU or Cholesky factorization needs to be done to solve the linear system \( (A - \lambda^{(k-1)} I) v = v^{(k-1)} \).

6 Computational cost

If \( A \) is an \( m \times m \) dense matrix, then the computational cost of the power iteration for each step is \( O(m^2) \) for multiplying a matrix with a vector. For
the inverse iteration, one LU factorization of $A - \mu I$ costs $O(m^3)$ flops, which only needs to be computed once in the inverse iteration, and then in each step of the iteration, the cost is $O(m^2)$ for the forward and backward substitutions. For the Rayleigh quotient iteration, the cost for each step is $O(m^3)$ for solving the system of linear equations.

The computation cost can be greatly reduced, especially for the Rayleigh quotient iteration, if we first reduce the symmetric matrix $A$ to a tridiagonal form by a sequence of symmetric orthogonal transformations which does not change the eigenvalues of $A$.

7 Reduction to Tridiagonal form

We know that for a real symmetric matrix $A$, it has an eigenvalue decomposition

$$A = Q\Lambda Q^T,$$

where $\Lambda$ is a diagonal matrix which contains the eigenvalues of $A$, and $Q$ is an orthogonal matrix which contains the eigenvectors of $A$.

In general it is impossible to obtain the eigenvalue decomposition by a direct method, as what was done to obtain the QR factorization or the LU factorization.

But we can be less ambitious. We can multiply a sequence of orthogonal matrices to the left and their transpose to the right of $A$ such that the result is a tridiagonal matrix. Such orthogonal matrices can be determined by the using Householder reflector.

By working on the first column/row of $A$ (excluding the first entry), we have

$$A 
\begin{bmatrix}
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times \\
\end{bmatrix}
\begin{bmatrix}
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times \\
\end{bmatrix}
\begin{bmatrix}
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times \\
\end{bmatrix}
\begin{bmatrix}
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times \\
\end{bmatrix}
\begin{bmatrix}
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times \\
\end{bmatrix}
\begin{bmatrix}
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times \\
\end{bmatrix}

Then work on the second column/row of $Q_1^T AQ_1$ (excluding the first two en-
tries), and we have

\[
\begin{pmatrix}
\times & \times \\
\times & \times & \times & \times \\
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times & \times \\
Q_1AQ_1^T
\end{pmatrix}
\Rightarrow
\begin{pmatrix}
\times & \times \\
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times & \times & \times \\
Q_2Q_1AQ_1^T
\end{pmatrix}
\Rightarrow
\begin{pmatrix}
\times & \times \\
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times & \times & \times \\
Q_2Q_1AQ_1^TQ_2^T
\end{pmatrix}
\]

Repeating the same process on the first \(m - 2\) columns/rows and we have at the end

\[
Q_{m-2}Q_{2}Q_1 A Q_1^T Q_2^T Q_{m-2}^T = T = \begin{bmatrix}
\times & \times & \times \\
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times & \times & \times \\
Q_{m-2}Q_1AQ_1^TQ_2^T
\end{bmatrix},
\]

where \(T\) is a tridiagonal matrix. We can see that \(A\) is similar to \(T\) and they have the same eigenvalues. The algorithm is shown on the next page.

In general, an eigenvalue problem solver first apply the above algorithm to reduce \(A\) into a tridiagonal form \(T\). Then an iteration, e.g., the power, the inverse, or the Rayleigh quotient, is implemented on \(T\) to obtain its eigenvalues, which are also eigenvalues of \(A\). After obtaining the eigenvalues, eigenvectors can be easily obtained by an applying inverse iteration.

The computational cost of the power, the inverse, and the Rayleigh quotient, iteration on a tridiagonal matrix of size \(m \times m\) is always \(O(m^2)\) in each iteration step.

**Algorithm: Reduction to tridiagonal form**

```
for k = 1 to m - 2
    x = A_{k+1:m,k}
    v_k = x + sign(x_1)\|x\|_2 e_1
    v_k = v_k / \|v_k\|_2
    A_{k+1:m,k:m} = A_{k+1:m,k:m} - 2v_k(v_k^T A_{k+1:m,k:m})
    A_{k:m,k+1:m} = A_{k:m,k+1:m} - 2(A_{k:m,k+1:m}v_k)v_k^T
end
```