The Power Method

- Label the eigenvalues in order of decreasing absolute value so $|\lambda_1| > |\lambda_2| > \ldots |\lambda_n|$.
- Consider the iteration formula:
  \[ y_{k+1} = Ay_k \]
  where we start with some initial $y_0$, so that:
  \[ y_k = A^ky_0 \]
- Then $y_k$ converges to the eigenvector $x_1$ corresponding the eigenvalue $\lambda_1$. 

Computational Methods
CMSC/AMSC/MAPL 460

Eigenvalues and Eigenvectors

Ramani Duraiswami,
Dept. of Computer Science
Eigen Decomposition

- Recall from previous class the Eigen decomposition
- Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues of the $n \times n$ matrix $A$ and $x_1, x_2, \ldots, x_n$ the corresponding eigenvectors.
- Let $\Lambda$ be the diagonal matrix with $\lambda_1, \lambda_2, \ldots, \lambda_n$ on the main diagonal.
- Let $X$ be the $n \times n$ matrix whose $j$th column is $x_j$.
- Then $AX = X \Lambda$, and so we have the eigen decomposition of $A$:
  $$A = X^t \Lambda X^{-1}$$
- This requires $X$ to be invertible, thus the eigenvectors of $A$ must be linearly independent.

Powers Method

- Also recall
- If $A = X^t \Lambda X^{-1}$ then:
  $$A^2 = (X^t \Lambda X^{-1}) (X^t \Lambda X^{-1}) = X^t \Lambda (X^{-1}X) \Lambda X^{-1} = X^t \Lambda^2 X^{-1}$$
  Hence we have:
  $$A^p = X^t \Lambda^p X^{-1}$$
- Thus, $A^p$ has the same eigenvectors as $A$, and its eigenvalues are $\lambda_1^p, \lambda_2^p, \ldots, \lambda_n^p$.
- We can use these results as the basis of an iterative algorithm for finding the eigenvalues of a matrix.
Proof of convergence of power method

• We know that $A^k = X \Lambda^k X^{-1}$, so:
  \[
y_k = A^k y_0 = X \Lambda^k y_0
  \]

• Now we have:
  \[
  \Lambda^k = \begin{pmatrix}
    \lambda_1^k \\
    \lambda_2^k \\
    \vdots \\
    \lambda_n^k
  \end{pmatrix} = \lambda_1^k 
  \]
  
• The terms on the diagonal get smaller in absolute value as $k$ increases, since $\lambda_1$ is the dominant eigenvalue.

Proof (continued)

• So we have
  \[
y_k = \lambda_1^k \begin{pmatrix}
    x_1 \\
    \vdots \\
    x_n
  \end{pmatrix} \begin{pmatrix}
    1 \\
    0 \\
    \vdots \\
    0
  \end{pmatrix} \begin{pmatrix}
    c_1 \\
    c_2 \\
    \vdots \\
    c_n
  \end{pmatrix} = \lambda_1^k c_1 x_1
  \]

• Since $\lambda_1^k c_1 x_1$ is just a constant times $x_1$ then we have the required result.
Example

- Let $A = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}$ and $y_0 = [1 \ 1]'$
- $y_1 = -4[2.50 \ 1.00]'$
- $y_2 = 10[2.80 \ 1.00]'$
- $y_3 = -22[2.91 \ 1.00]'$
- $y_4 = 46[2.96 \ 1.00]'$
- $y_5 = -94[2.98 \ 1.00]'$
- $y_6 = -190[2.99 \ 1.00]'$
- The iteration is converging on a scalar multiple of $[3 \ 1]'$, which is the correct dominant eigenvector.

Rayleigh Quotient

- Note that once we have the eigenvector, the corresponding eigenvalue can be obtained from the Rayleigh quotient:
  \[
  \frac{\text{dot}(Ax, x)}{\text{dot}(x, x)}
  \]
  where $\text{dot}(a, b)$ is the scalar product of vectors $a$ and $b$ defined by:
  \[
  \text{dot}(a, b) = a_1b_1 + a_2b_2 + \ldots + a_nb_n
  \]
- So for our example, $\lambda_1 = -2$. 
Scaling

- The $\lambda_1^k$ can cause problems as it may become very large as the iteration progresses.
- To avoid this problem we scale the iteration formula:

$$y_{k+1} = A(y_k / r_{k+1})$$

where $r_{k+1}$ is the component of $Ay_k$ with largest absolute value.

Example with Scaling

- Let $A = [2 -12; 1 -5]$ and $y_0 = [1 1]'$
- $Ay_0 = [-10 -4]'$ so $r_1 = -10$ and $y_1 = [1.00 0.40]'$.
- $Ay_1 = [-2.8 -1.0]'$ so $r_2 = -2.8$ and $y_2 = [1.0 0.3571]'$.
- $Ay_2 = [2.2857 -0.757]'$ so $r_3 = -2.2857$ and $y_3 = [1.0 0.3437]'$.
- $Ay_3 = [-2.1250 -0.7187]'$ so $r_4 = -2.1250$ and $y_4 = [1.0 0.3382]'$.
- $Ay_4 = [-2.0588 -0.6912]'$ so $r_5 = -2.0588$ and $y_5 = [1.0 0.3357]'$.
- $Ay_5 = [-2.0286 -0.6786]'$ so $r_6 = -2.0286$ and $y_6 = [1.0 0.3345]'$.
- $r$ is converging to the correct eigenvalue -2.
Scaling Factor

- At step $k+1$, the scaling factor $r_{k+1}$ is the component with largest absolute value is $A\mathbf{y}_k$.
- When $k$ is sufficiently large $A\mathbf{y}_k \approx \lambda_1 \mathbf{y}_k$.
- The component with largest absolute value in $\lambda_1 \mathbf{y}_k$ is $\lambda_1$ (since $\mathbf{y}_k$ was scaled in the previous step to have largest component 1).
- Hence, $r_{k+1} \rightarrow \lambda_1$ as $k \rightarrow \infty$.

MATLAB Code

```matlab
function [lambda,y]=powerMethod(A,y,n)
for (i=1:n)
    y = A*y;
    [c j] = max(abs(y));
    lambda = y(j);
    y = y/lambda;
end
```
Convergence

- The Power Method relies on us being able to ignore terms of the form $(\lambda_i / \lambda_1)^k$ when $k$ is large enough.
- Thus, the convergence of the Power Method depends on $|\lambda_2|/|\lambda_1|$.
- If $|\lambda_2|/|\lambda_1|=1$ the method will not converge.
- If $|\lambda_2|/|\lambda_1|$ is close to 1 the method will converge slowly.

Similar Matrices

- Two matrices $A$ and $B$ are said to be similar it is possible to relate them as
  
  $B = T^{-1}AT$  \hspace{1cm}  $TBT^{-1}=A$

- Here $T$ is any non singular matrix, which is the similarity transform matrix

- **Theorem:** Similar matrices have the same eigenvalues and their eigen-vectors are related via the similarity transform.

- **Proof.** Let $(x, \lambda)$ be an eigen-pair for $A$. Then $Ax=\lambda x$.
  
  Let $y=T^{-1}x$ and $x=Ty$
  
  Then $Ax=\lambda x$.
  
  Premultiply by $T^{-1}$ to get $T^{-1}ATT^{-1}x = \lambda T^{-1}x$
  
  So $By=\lambda y$
The QR Algorithm

• The QR algorithm for finding eigenvalues is based on the QR factorisation we learnt in the least squares part of the course.
• Recall the QR factorization represents a matrix $A$ as:

$$A = QR$$

where $Q$ is a matrix whose columns are orthonormal, and $R$ is an upper triangular matrix.
• Recall that $Q^HQ = I$ and $Q^{-1} = Q^H$.
• $Q$ is termed a unitary matrix.

QR Algorithm without Shifts

\[
\begin{align*}
A_0 &= A \\
&\text{for } k=1,2,\ldots \\
Q_k R_k &= A_k \\
A_{k+1} &= R_k Q_k \\
\text{end}
\end{align*}
\]

Since:

$$A_{k+1} = R_k Q_k = Q_k^{-1} A_k Q_k$$

then $A_k$ and $A_{k+1}$ are similar and so have the same eigenvalues.

$A_{k+1}$ tends to an upper triangular matrix with the same eigenvalues as $A$. These eigenvalues lie along the main diagonal of $A_{k+1}$. 
QR Algorithm with Shift

\[
A_0 = A \\
\text{for } k=1,2,\ldots \\
s = A_k(n,n) \\
Q_k R_k = A_k - sI \\
A_{k+1} = R_k Q_k + sI \\
\text{end}
\]

Since:

\[
A_{k+1} = R_k Q_k + sI \\
= Q_k^{-1}(A_k - sI)Q_k + sI \\
= Q_k^{-1}A_k Q_k
\]

so once again \(A_k\) and \(A_{k+1}\) are similar and so have the same eigenvalues.

The shift operation subtracts \(s\) from each eigenvalue of \(A\), and speeds up convergence.

MATLAB Code for QR Algorithm

- Let \(A\) be an \(n \times n\) matrix

\[
n = \text{size}(A,1); \\
I = \text{eye}(n,n); \\
s = A(n,n); \ [Q,R] = \text{qr}(A-s*I); \ A = R*Q+s*I
\]

- Use the up arrow key in MATLAB to iterate or put a loop round the last line.
The SVD

• **Definition:** Every matrix $A$ of dimensions $m \times n$ ($m \geq n$) can be decomposed as
  $$A = U \Sigma V^*$$

• where
  - $U$ has dimension $m \times m$ and $U^*U = I$,
  - $\Sigma$ has dimension $m \times n$,
    the only nonzeros are on the main diagonal, and they are nonnegative real numbers
    $$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n,$$
  - $V$ has dimension $n \times n$ and $V^* V = I$.

Relation with the Eigenvalue Decomposition

• Let $A = U \Sigma V^*$ . Then
  $$A^* A = (U\Sigma V^*)^* U \Sigma V^*$$
  $$= V \Sigma^* U^* U \Sigma V^* = V \Sigma^2 V^*$$

• This tells us that the singular value decomposition of $A$ is related to the Eigenvalue decomposition of $A^* A$

• Recall eigen value decomposition $A = (X A X^*)$
  - So $V$ which contains the right singular vectors of $A$ has the right eigenvectors of $A^* A$
  $\square \Sigma^2$ are the eigenvalues of $A^* A$
  - The **singular values** $\sigma_i$ of $A$ are the square roots of the **eigenvalues** of $A^* A$. 