Spectral Theorem.

For real symmetric matrix, A, there exists a set of unit vectors $u_1, \ldots, u_n$, and real numbers λ_i such $u_i \perp u_j$ and $Au_i = \lambda_i u_i$.

Spectral Theorem.

For real symmetric matrix, A, there exists a set of unit vectors $u_1, \ldots, u_n$, and real numbers λ_i such $u_i \perp u_j$ and $Au_i = \lambda_i u_i$.

One idea in proof:

Spectral Theorem.

For real symmetric matrix, A, there exists a set of unit vectors $u_1, \ldots, u_n$, and real numbers λ_i such $u_i \perp u_j$ and $Au_i = \lambda_i u_i$.

One idea in proof:

$\min x^T Ax$ s.t $\|x\| = 1$.

Spectral Theorem.

For real symmetric matrix, A, there exists a set of unit vectors $u_1, \ldots, u_n$, and real numbers λ_i such $u_i \perp u_j$ and $Au_i = \lambda_i u_i$.

One idea in proof:

$$\min x^T A x \text{ s.t } \|x\| = 1.$$

How?
 Constrained optimization.

Spectral Theorem.

For real symmetric matrix, A, there exists a set of unit vectors $u_1,\ldots,u_n$, and real numbers λ_i such $u_i \perp u_j$ and $Au_i = \lambda_i u_i$.

One idea in proof:

$\min x^T Ax$ s.t $\|x\| = 1$.

How?
Constrained optimization.

$f(x) = x^T Ax - \lambda(\|x\|^2 - 1)$

Spectral Theorem.

For real symmetric matrix, A, there exists a set of unit vectors $u_1, \ldots, u_n$, and real numbers λ_i such $u_i \perp u_j$ and $Au_i = \lambda_i u_i$.

One idea in proof:

$$\min x^T Ax \text{ s.t } \|x\| = 1.$$

How?

Constrained optimization.

$$f(x) = x^T Ax - \lambda(\|x\|^2 - 1)$$

Fix λ.

Spectral Theorem.

For real symmetric matrix, A, there exists a set of unit vectors $u_1, \ldots, u_n$, and real numbers λ_i such $u_i \perp u_j$ and $Au_i = \lambda_i u_i$.

One idea in proof:

$\min x^T Ax$ s.t $\|x\| = 1$.

How?
Constrained optimization.

$f(x) = x^T Ax - \lambda(\|x\|^2 - 1)$

Fix λ. Minimize for x.

Spectral Theorem.

For real symmetric matrix, A, there exists a set of unit vectors $u_1, \ldots, u_n$, and real numbers λ_i such $u_i \perp u_j$ and $Au_i = \lambda_i u_i$.

One idea in proof:

$$\min x^T Ax \text{ s.t } \|x\| = 1.$$

How?

Constrained optimization.

$$f(x) = x^T Ax - \lambda(\|x\|^2 - 1)$$

Fix λ. Minimize for x.

$$\nabla f(x) = 2Ax - 2\lambda x = 0$$

Spectral Theorem.

For real symmetric matrix, A, there exists a set of unit vectors $u_1, \ldots, u_n$, and real numbers λ_i such $u_i \perp u_j$ and $Au_i = \lambda_i u_i$.

One idea in proof:

$$\min x^T Ax \text{ s.t } \|x\| = 1.$$

How?
Constrained optimization.

$$f(x) = x^T Ax - \lambda(\|x\|^2 - 1)$$

Fix λ. Minimize for x.
$\nabla f(x) = 2Ax - 2\lambda x = 0 \implies$ Find when: $Ax = \lambda x$.

Spectral Theorem.

For real symmetric matrix, A, there exists a set of unit vectors $u_1,\ldots,u_n$, and real numbers λ_i such $u_i \perp u_j$ and $Au_i = \lambda_i u_i$.

One idea in proof:

$\min x^T Ax$ s.t $\|x\| = 1$.

How?
Constrained optimization.

$f(x) = x^T Ax - \lambda(\|x\|^2 - 1)$

Fix λ. Minimize for x.
$\nabla f(x) = 2Ax - 2\lambda x = 0 \implies$ Find when: $Ax = \lambda x$.

Minimizer is “eigenvector”.

Finding an eigenvector: power method

A has eigenpairs: $(u_1, \lambda_1)...(u_n, \lambda_n)$.

Finding an eigenvector: power method

A has eigenpairs: $(u_1, \lambda_1) ... (u_n, \lambda_n)$.

$Au_i = \lambda_i u_i$

Finding an eigenvector: power method

A has eigenpairs: $(u_1, \lambda_1)...(u_n, \lambda_n)$.

$Au_i = \lambda_i u_i$
$u_i \perp u_j$

Finding an eigenvector: power method

A has eigenpairs: $(u_1, \lambda_1)...(u_n, \lambda_n)$.

$Au_i = \lambda_i u_i$
$u_i \perp u_j$
$\lambda_1 > \lambda_2 ... > \lambda_n$

Finding an eigenvector: power method

A has eigenpairs: $(u_1, \lambda_1) \ldots (u_n, \lambda_n)$.

$Au_i = \lambda_i u_i$
$u_i \perp u_j$
$\lambda_1 > \lambda_2 \ldots > \lambda_n$

Take "random" x_0.

Finding an eigenvector: power method

A has eigenpairs: $(u_1, \lambda_1)...(u_n, \lambda_n)$.

$Au_i = \lambda_i u_i$
$u_i \perp u_j$
$\lambda_1 > \lambda_2 ... > \lambda_n$

Take "random" x_0.

$x_0 = a_1 u_1 + \ldots a_2 u_2 + \cdots + a_n u_n$.

Finding an eigenvector: power method

A has eigenpairs: $(u_1, \lambda_1)...(u_n, \lambda_n)$.

$$Au_i = \lambda_i u_i$$
$$u_i \perp u_j$$
$$\lambda_1 > \lambda_2 ... > \lambda_n$$

Take "random" x_0.

$$x_0 = a_1 u_1 + \ldots a_2 u_2 + \cdots + a_n u_n.$$

$$x_{t+1} = Ax_t = A^t x_0$$

Finding an eigenvector: power method

A has eigenpairs: $(u_1, \lambda_1)...(u_n, \lambda_n)$.

$$Au_i = \lambda_i u_i$$
$$u_i \perp u_j$$
$$\lambda_1 > \lambda_2 ... > \lambda_n$$

Take “random” x_0.

$$x_0 = a_1 u_1 + \ldots a_2 u_2 + \cdots + a_n u_n.$$

$$x_{t+1} = Ax_t = A^t x_0$$

$$x_{t+1} = \lambda_1^t a_1 u_1 + \cdots \lambda_n^t a_n u_n$$

Finding an eigenvector: power method

A has eigenpairs: $(u_1, \lambda_1)...(u_n, \lambda_n)$.

$Au_i = \lambda_i u_i$
$u_i \perp u_j$
$\lambda_1 > \lambda_2 ... > \lambda_n$

Take "random" x_0.

$x_0 = a_1 u_1 + \ldots a_2 u_2 + \cdots + a_n u_n$.

$x_{t+1} = Ax_t = A^t x_0$

$x_{t+1} = \lambda_1^t a_1 u_1 + \cdots \lambda_n^t a_n u_n$

Since $\lambda_1 > \lambda 2 > \ldots$.

Finding an eigenvector: power method

A has eigenpairs: $(u_1, \lambda_1)...(u_n, \lambda_n)$.

$Au_i = \lambda_i u_i$
$u_i \perp u_j$
$\lambda_1 > \lambda_2 ... > \lambda_n$

Take "random" x_0.

$x_0 = a_1 u_1 + \ldots a_2 u_2 + \cdots + a_n u_n$.

$x_{t+1} = Ax_t = A^t x_0$

$x_{t+1} = \lambda_1^t a_1 u_1 + \cdots \lambda_n^t a_n u_n$

Since $\lambda_1 > \lambda 2 > \ldots$.

$x_t / |x_t|$ converges to u_1.

Finding an eigenvector: power method

A has eigenpairs: $(u_1, \lambda_1)...(u_n, \lambda_n)$.

$Au_i = \lambda_i u_i$
$u_i \perp u_j$
$\lambda_1 > \lambda_2 ... > \lambda_n$

Take "random" x_0.

$x_0 = a_1 u_1 + \ldots a_2 u_2 + \cdots + a_n u_n$.

$x_{t+1} = Ax_t = A^t x_0$

$x_{t+1} = \lambda_1^t a_1 u_1 + \cdots \lambda_n^t a_n u_n$

Since $\lambda_1 > \lambda 2 > \ldots$.

$x_t / |x_t|$ converges to u_1.

Get rest?

Finding an eigenvector: power method

A has eigenpairs: $(u_1, \lambda_1) ... (u_n, \lambda_n)$.

$Au_i = \lambda_i u_i$
$u_i \perp u_j$
$\lambda_1 > \lambda_2 ... > \lambda_n$

Take "random" x_0.

$x_0 = a_1 u_1 + \ldots a_2 u_2 + \cdots + a_n u_n$.

$x_{t+1} = Ax_t = A^t x_0$

$x_{t+1} = \lambda_1^t a_1 u_1 + \cdots \lambda_n^t a_n u_n$

Since $\lambda_1 > \lambda 2 > \ldots$.

$x_t / |x_t|$ converges to u_1.

Get rest? Orthoganalize and induction.

Today.

Linear Solvers.

Today.

Linear Solvers.

A little background, intuition, alternative.

Classical Stuff.

$Ax = b$?

Classical Stuff.

$Ax = b$?

$x_{n+1} = x_n + \alpha(b - Ax)$.

Classical Stuff.

$Ax = b$?

$x_{n+1} = x_n + \alpha(b - Ax)$.

Intuition: $\frac{1}{2}x^T Ax - bx$. Gradient: $Ax - b$.

Classical Stuff.

$Ax = b$?

$x_{n+1} = x_n + \alpha(b - Ax)$.

Intuition: $\frac{1}{2}x^T Ax - bx$.   Gradient: $Ax - b$.

Classical Stuff.

$Ax = b$?

$x_{n+1} = x_n + \alpha(b - Ax)$.

Intuition: $\frac{1}{2}x^T Ax - bx$. Gradient: $Ax - b$.

(1) Critical point: $Ax = b$.

Classical Stuff.

$Ax = b$?

$x_{n+1} = x_n + \alpha(b - Ax)$.

Intuition: $\frac{1}{2}x^T Ax - bx$. Gradient: $Ax - b$.

(1) Critical point: $Ax = b$.

(2) Gradient step: $x_n - \alpha \nabla f(x) = x_n + \alpha(b - Ax)$.

Classical Stuff.

$Ax = b$?

$x_{n+1} = x_n + \alpha(b - Ax)$.

Intuition: $\frac{1}{2}x^T Ax - bx$. Gradient: $Ax - b$.

(1) Critical point: $Ax = b$.

(2) Gradient step: $x_n - \alpha \nabla f(x) = x_n + \alpha(b - Ax)$.

Could analyse using gradient descent (lipshitz constant?)

Classical Stuff.

$Ax = b$?

$x_{n+1} = x_n + \alpha(b - Ax)$.

Intuition: $\frac{1}{2}x^T Ax - bx$.   Gradient: $Ax - b$.

(1) Critical point: $Ax = b$.

(2) Gradient step: $x_n - \alpha \nabla f(x) = x_n + \alpha(b - Ax)$.

Could analyse using gradient descent (lipshitz constant?) or Accelerated Gradient Descent.

Classical Stuff.

$Ax = b$?

$x_{n+1} = x_n + \alpha(b - Ax)$.

Intuition: $\frac{1}{2}x^T Ax - bx$. Gradient: $Ax - b$.

(1) Critical point: $Ax = b$.

(2) Gradient step: $x_n - \alpha \nabla f(x) = x_n + \alpha(b - Ax)$.

Could analyse using gradient descent (lipshitz constant?) or Accelerated Gradient Descent.

E.g., Lipshitz constant has to do with A.

Classical Stuff.

$Ax = b$?

$x_{n+1} = x_n + \alpha(b - Ax)$.

Intuition: $\frac{1}{2}x^T Ax - bx$. Gradient: $Ax - b$.

(1) Critical point: $Ax = b$.

(2) Gradient step: $x_n - \alpha\nabla f(x) = x_n + \alpha(b - Ax)$.

Could analyse using gradient descent (lipshitz constant?) or Accelerated Gradient Descent.

E.g., Lipshitz constant has to do with A.
Ratio of largest to smallest eigenvalues.

Classical Stuff.

$Ax = b$?

$x_{n+1} = x_n + \alpha(b - Ax)$.

Intuition: $\frac{1}{2}x^T Ax - bx$. Gradient: $Ax - b$.

(1) Critical point: $Ax = b$.

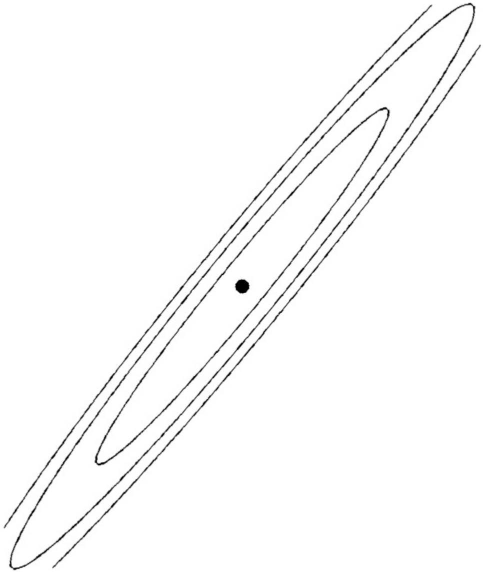
(2) Gradient step: $x_n - \alpha\nabla f(x) = x_n + \alpha(b - Ax)$.

Could analyse using gradient descent (lipshitz constant?)
or Accelerated Gradient Descent.

E.g., Lipshitz constant has to do with A.
Ratio of largest to smallest eigenvalues.

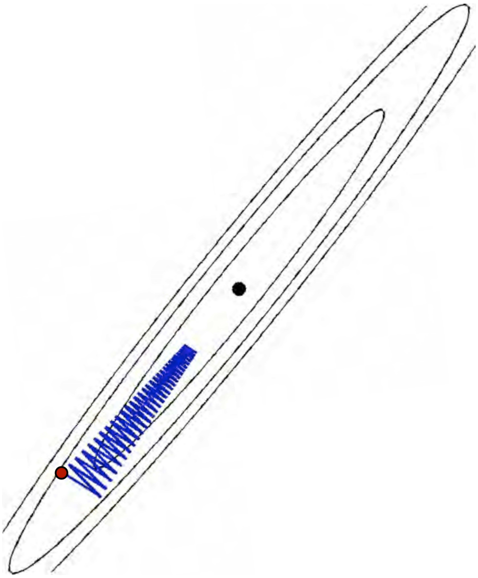
Linear Coupling

Momentum View:



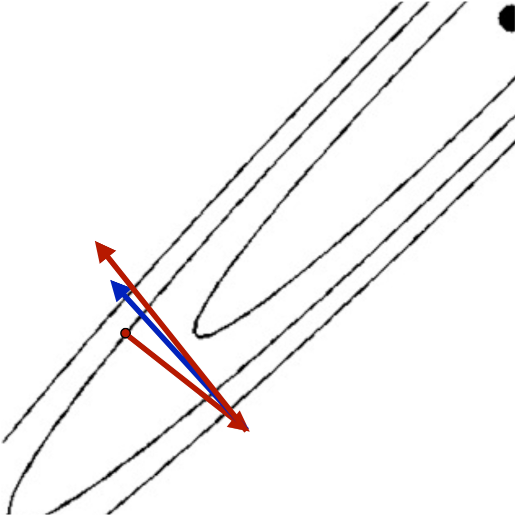
Linear Coupling

Momentum View:



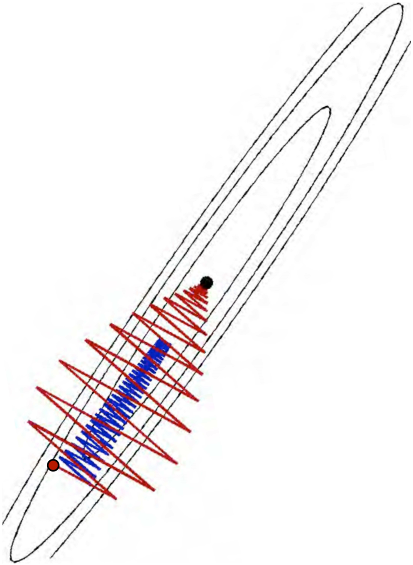
Linear Coupling

Momentum View:



Linear Coupling

Momentum View:



Eigenvalue Analysis.

$Ax = b$?

Eigenvalue Analysis.

$Ax = b$?

$x_{n+1} = x_n + \alpha(b - Ax)$.

Eigenvalue Analysis.

$Ax = b$?

$x_{n+1} = x_n + \alpha(b - Ax)$.

Convergence:

Eigenvalue Analysis.

$Ax = b$?

$x_{n+1} = x_n + \alpha(b - Ax)$.

Convergence:

$|x_{n+1} - x^*|^2 = |x_n + \alpha(b - Ax_n) - x^*|$

Eigenvalue Analysis.

$Ax = b$?

$x_{n+1} = x_n + \alpha(b - Ax)$.

Convergence:

$$|x_{n+1} - x^*|^2 = |x_n + \alpha(b - Ax_n) - x^*|$$
$$= \|x_n - x^* - \alpha A(x_n - x^*)\| \quad \text{From } b = Ax^*.$$

Eigenvalue Analysis.

$Ax = b$?

$x_{n+1} = x_n + \alpha(b - Ax)$.

Convergence:

$$|x_{n+1} - x^*|^2 = |x_n + \alpha(b - Ax_n) - x^*|$$
$$= \|x_n - x^* - \alpha A(x_n - x^*)\| \quad \text{From } b = Ax^*.$$

Eigenvectors decomposition: $(x^* - x_n) = v = a_1 v_1 + \cdots a_n v_n$

Eigenvalue Analysis.

$Ax = b$?

$x_{n+1} = x_n + \alpha(b - Ax)$.

Convergence:

$$|x_{n+1} - x^*|^2 = |x_n + \alpha(b - Ax_n) - x^*|$$
$$= \|x_n - x^* - \alpha A(x_n - x^*)\| \quad \text{From } b = Ax^*.$$

Eigenvectors decomposition: $(x^* - x_n) = v = a_1 v_1 + \cdots a_n v_n$
with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$.

Eigenvalue Analysis.

$Ax = b$?

$x_{n+1} = x_n + \alpha(b - Ax)$.

Convergence:

$|x_{n+1} - x^*|^2 = |x_n + \alpha(b - Ax_n) - x^*|$

$= \|x_n - x^* - \alpha A(x_n - x^*)\|$ From $b = Ax^*$.

Eigenvectors decomposition: $(x^* - x_n) = v = a_1 v_1 + \cdots a_n v_n$

with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$.

Notice $|v|^2 = \sum_i a_i^2$.

Eigenvalue Analysis.

$Ax = b$?

$x_{n+1} = x_n + \alpha(b - Ax)$.

Convergence:

$|x_{n+1} - x^*|^2 = |x_n + \alpha(b - Ax_n) - x^*|$

$= \|x_n - x^* - \alpha A(x_n - x^*)\|$ From $b = Ax^*$.

Eigenvectors decomposition: $(x^* - x_n) = v = a_1 v_1 + \cdots a_n v_n$

with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$.

Notice $|v|^2 = \sum_i a_i^2$.

and $Av = \lambda_1 a_1 v_1 + \cdots \lambda_n a_n v_n$

Eigenvalue Analysis.

$Ax = b$?

$x_{n+1} = x_n + \alpha(b - Ax)$.

Convergence:

$|x_{n+1} - x^*|^2 = |x_n + \alpha(b - Ax_n) - x^*|$

$= \|x_n - x^* - \alpha A(x_n - x^*)\|$ From $b = Ax^*$.

Eigenvectors decomposition: $(x^* - x_n) = v = a_1 v_1 + \cdots a_n v_n$

with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$.

Notice $|v|^2 = \sum_i a_i^2$.

and $Av = \lambda_1 a_1 v_1 + \cdots \lambda_n a_n v_n$

$\implies |x_{n+1} - x^*|^2 = \sum_i (1 - \alpha\lambda_i)^2 a_i^2$.

Eigenvalue Analysis.

$Ax = b$?

$x_{n+1} = x_n + \alpha(b - Ax)$.

Convergence:

$|x_{n+1} - x^*|^2 = |x_n + \alpha(b - Ax_n) - x^*|$

$= \|x_n - x^* - \alpha A(x_n - x^*)\|$ From $b = Ax^*$.

Eigenvectors decomposition: $(x^* - x_n) = v = a_1 v_1 + \cdots a_n v_n$

with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$.

Notice $|v|^2 = \sum_i a_i^2$.

and $Av = \lambda_1 a_1 v_1 + \cdots \lambda_n a_n v_n$

$\implies |x_{n+1} - x^*|^2 = \sum_i (1 - \alpha\lambda_i)^2 a_i^2$.

Eigenvalue Analysis.

$Ax = b$?

$x_{n+1} = x_n + \alpha(b - Ax)$.

Convergence:

$|x_{n+1} - x^*|^2 = |x_n + \alpha(b - Ax_n) - x^*|$
$= \|x_n - x^* - \alpha A(x_n - x^*)\|$ From $b = Ax^*$.

Eigenvectors decomposition: $(x^* - x_n) = v = a_1 v_1 + \cdots a_n v_n$
with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$.

Notice $|v|^2 = \sum_i a_i^2$.
and $Av = \lambda_1 a_1 v_1 + \cdots \lambda_n a_n v_n$
$\implies |x_{n+1} - x^*|^2 = \sum_i (1 - \alpha\lambda_i)^2 a_i^2$.

Setting $\alpha = 2/(\lambda_1 + \lambda_2)$,

Eigenvalue Analysis.

$Ax = b$?

$x_{n+1} = x_n + \alpha(b - Ax)$.

Convergence:

$|x_{n+1} - x^*|^2 = |x_n + \alpha(b - Ax_n) - x^*|$

$= \|x_n - x^* - \alpha A(x_n - x^*)\|$ From $b = Ax^*$.

Eigenvectors decomposition: $(x^* - x_n) = v = a_1 v_1 + \cdots a_n v_n$

with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$.

Notice $|v|^2 = \sum_i a_i^2$.

and $Av = \lambda_1 a_1 v_1 + \cdots \lambda_n a_n v_n$

$\implies |x_{n+1} - x^*|^2 = \sum_i (1 - \alpha\lambda_i)^2 a_i^2$.

Setting $\alpha = 2/(\lambda_1 + \lambda_2)$, and some math (e.g., Taylors) yields

Eigenvalue Analysis.

$Ax = b$?

$x_{n+1} = x_n + \alpha(b - Ax)$.

Convergence:

$|x_{n+1} - x^*|^2 = |x_n + \alpha(b - Ax_n) - x^*|$

$= \|x_n - x^* - \alpha A(x_n - x^*)\|$ From $b = Ax^*$.

Eigenvectors decomposition: $(x^* - x_n) = v = a_1 v_1 + \cdots a_n v_n$

with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$.

Notice $|v|^2 = \sum_i a_i^2$.

and $Av = \lambda_1 a_1 v_1 + \cdots \lambda_n a_n v_n$

$\implies |x_{n+1} - x^*|^2 = \sum_i (1 - \alpha\lambda_i)^2 a_i^2$.

Setting $\alpha = 2/(\lambda_1 + \lambda_2)$, and some math (e.g., Taylors) yields

$(1 - \alpha\lambda_i)^2 \leq (1 - \Omega(\lambda_n/\lambda_1))^2$

Eigenvalue Analysis.

$Ax = b$?

$x_{n+1} = x_n + \alpha(b - Ax)$.

Convergence:
$|x_{n+1} - x^*|^2 = |x_n + \alpha(b - Ax_n) - x^*|$
$= \|x_n - x^* - \alpha A(x_n - x^*)\|$ From $b = Ax^*$.

Eigenvectors decomposition: $(x^* - x_n) = v = a_1 v_1 + \cdots a_n v_n$
with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$.

Notice $|v|^2 = \sum_i a_i^2$.
and $Av = \lambda_1 a_1 v_1 + \cdots \lambda_n a_n v_n$
$\implies |x_{n+1} - x^*|^2 = \sum_i (1 - \alpha\lambda_i)^2 a_i^2$.

Setting $\alpha = 2/(\lambda_1 + \lambda_2)$, and some math (e.g., Taylors) yields
$(1 - \alpha\lambda_i)^2 \leq (1 - \Omega(\lambda_n/\lambda_1))^2$

$\implies O(\lambda_1/\lambda_n)$ iterations halves the size of the error.

Eigenvalue Analysis.

$Ax = b$?

$x_{n+1} = x_n + \alpha(b - Ax)$.

Convergence:

$$|x_{n+1} - x^*|^2 = |x_n + \alpha(b - Ax_n) - x^*|$$
$$= \|x_n - x^* - \alpha A(x_n - x^*)\| \quad \text{From } b = Ax^*.$$

Eigenvectors decomposition: $(x^* - x_n) = v = a_1 v_1 + \cdots a_n v_n$ with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$.

Notice $|v|^2 = \sum_i a_i^2$.

and $Av = \lambda_1 a_1 v_1 + \cdots \lambda_n a_n v_n$

$$\implies |x_{n+1} - x^*|^2 = \sum_i (1 - \alpha\lambda_i)^2 a_i^2.$$

Setting $\alpha = 2/(\lambda_1 + \lambda_2)$, and some math (e.g., Taylors) yields

$$(1 - \alpha\lambda_i)^2 \leq (1 - \Omega(\lambda_n/\lambda_1))^2$$

$\implies O(\lambda_1/\lambda_n)$ iterations halves the size of the error.

Note: $\log \frac{1}{\varepsilon}$ dependence on error ε.

Eigenvalue Analysis.

$Ax = b$?

$x_{n+1} = x_n + \alpha(b - Ax)$.

Convergence:

$|x_{n+1} - x^*|^2 = |x_n + \alpha(b - Ax_n) - x^*|$

$= \|x_n - x^* - \alpha A(x_n - x^*)\|$ From $b = Ax^*$.

Eigenvectors decomposition: $(x^* - x_n) = v = a_1 v_1 + \cdots a_n v_n$

with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$.

Notice $|v|^2 = \sum_i a_i^2$.

and $Av = \lambda_1 a_1 v_1 + \cdots \lambda_n a_n v_n$

$\implies |x_{n+1} - x^*|^2 = \sum_i (1 - \alpha\lambda_i)^2 a_i^2$.

Setting $\alpha = 2/(\lambda_1 + \lambda_2)$, and some math (e.g., Taylors) yields

$(1 - \alpha\lambda_i)^2 \leq (1 - \Omega(\lambda_n/\lambda_1))^2$

$\implies O(\lambda_1/\lambda_n)$ iterations halves the size of the error.

Note: $\log \frac{1}{\varepsilon}$ dependence on error ε.

Can do $\sqrt{\lambda_1/\lambda_n}$ using Chebyshev or Conjugate Gradient or Accelerated Gradient Descent)

Eigenvalue Analysis.

$Ax = b$?

$x_{n+1} = x_n + \alpha(b - Ax)$.

Convergence:

$$|x_{n+1} - x^*|^2 = |x_n + \alpha(b - Ax_n) - x^*|$$
$$= \|x_n - x^* - \alpha A(x_n - x^*)\| \quad \text{From } b = Ax^*.$$

Eigenvectors decomposition: $(x^* - x_n) = v = a_1 v_1 + \cdots a_n v_n$
with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$.

Notice $|v|^2 = \sum_i a_i^2$.
and $Av = \lambda_1 a_1 v_1 + \cdots \lambda_n a_n v_n$
$\implies |x_{n+1} - x^*|^2 = \sum_i (1 - \alpha\lambda_i)^2 a_i^2$.

Setting $\alpha = 2/(\lambda_1 + \lambda_2)$, and some math (e.g., Taylors) yields
$$(1 - \alpha\lambda_i)^2 \leq (1 - \Omega(\lambda_n/\lambda_1))^2$$

$\implies O(\lambda_1/\lambda_n)$ iterations halves the size of the error.
Note: $\log \frac{1}{\varepsilon}$ dependence on error ε.

Can do $\sqrt{\lambda_1/\lambda_n}$ using Chebyshev or Conjugate Gradient or Accelerated Gradient Descent)

Recall: λ_1 is large eigenvalue, λ_n is small.

Graph Laplacians.

$G = (V, E)$,

Graph Laplacians.

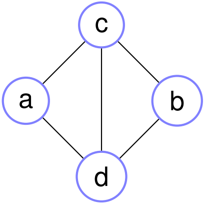
$$G = (V, E),$$

$$L(G) = L = dI - A.$$

Graph Laplacians.

$G = (V, E)$,

$L(G) = L = dI - A$.



	a	b	c	d
a	2	-1	-1	0
b	-1	2	0	-1
c	-1	-1	3	-1
d	-1	-1	-1	3

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$.

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

$\frac{x^T Ax}{d|x|^2}$ is proxy for "eigenvalue".

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

- $dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.
- $\frac{x^T A x}{d|x|^2}$ is proxy for "eigenvalue".
  - $x^T(dI - A)x = d|x|^2 - x^T A x$.

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

$\frac{x^T Ax}{d|x|^2}$ is proxy for "eigenvalue".

$x^T(dI - A)x = d|x|^2 - x^T Ax$.

If different for different x, then λ_1/λ_n is large:

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

$\frac{x^T A x}{d|x|^2}$ is proxy for "eigenvalue".

$x^T(dI - A)x = d|x|^2 - x^T A x$.

If different for different x, then λ_1/λ_n is large:

$Av_i = \lambda_i v_i \rightarrow \frac{v_i^T A v_i}{|v_i|^2} = \lambda_i$.

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

$\frac{x^T A x}{d|x|^2}$ is proxy for "eigenvalue".

$x^T(dI - A)x = d|x|^2 - x^T A x$.

If different for different x, then λ_1/λ_n is large:

$Av_i = \lambda_i v_i \rightarrow \frac{v_i^T A v_i}{|v_i|^2} = \lambda_i$.

Expander Graph:

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

$\frac{x^T Ax}{d|x|^2}$ is proxy for "eigenvalue".

$x^T(dI - A)x = d|x|^2 - x^T Ax$.

If different for different x, then λ_1/λ_n is large:

$Av_i = \lambda_i v_i \rightarrow \frac{v_i^T Av_i}{|v_i|^2} = \lambda_i$.

Expander Graph:

All $x \perp \bar{\mathbf{1}}$: $\sum_i x_i = 0$.

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

$\frac{x^T A x}{d|x|^2}$ is proxy for "eigenvalue".

$x^T(dI - A)x = d|x|^2 - x^T A x$.

If different for different x, then λ_1/λ_n is large:

$Av_i = \lambda_i v_i \rightarrow \frac{v_i^T A v_i}{|v_i|^2} = \lambda_i$.

Expander Graph:

All $x \perp \bar{\mathbf{1}}$: $\sum_i x_i = 0$.

Degree d, think of random ± 1 labels.

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

$\frac{x^T A x}{d|x|^2}$ is proxy for "eigenvalue".

$x^T(dI - A)x = d|x|^2 - x^T A x$.

If different for different x, then λ_1/λ_n is large:

$Av_i = \lambda_i v_i \to \frac{v_i^T A v_i}{|v_i|^2} = \lambda_i$.

Expander Graph:

All $x \perp \bar{\mathbf{1}}$: $\sum_i x_i = 0$.

Degree d, think of random ± 1 labels.

$(Ax)_i$ = average value of labels.

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

$\frac{x^T Ax}{d|x|^2}$ is proxy for "eigenvalue".

$x^T(dI - A)x = d|x|^2 - x^T Ax$.

If different for different x, then λ_1/λ_n is large:

$Av_i = \lambda_i v_i \rightarrow \frac{v_i^T Av_i}{|v_i|^2} = \lambda_i$.

Expander Graph:

All $x \perp \bar{\mathbf{1}}$: $\sum_i x_i = 0$.

Degree d, think of random ± 1 labels.

$(Ax)_i =$ average value of labels.

Roughly $\pm\sqrt{d}$.

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

$\frac{x^T Ax}{d|x|^2}$ is proxy for "eigenvalue".

$x^T(dI - A)x = d|x|^2 - x^T Ax$.

If different for different x, then λ_1/λ_n is large:

$Av_i = \lambda_i v_i \rightarrow \frac{v_i^T Av_i}{|v_i|^2} = \lambda_i$.

Expander Graph:

All $x \perp \bar{\mathbf{1}}$: $\sum_i x_i = 0$.

Degree d, think of random ± 1 labels.

$(Ax)_i =$ average value of labels.

Roughly $\pm\sqrt{d}$.

Expander: "no good cuts"

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

$\frac{x^T Ax}{d|x|^2}$ is proxy for "eigenvalue".

$x^T(dI - A)x = d|x|^2 - x^T Ax$.

If different for different x, then λ_1/λ_n is large:

$Av_i = \lambda_i v_i \rightarrow \frac{v_i^T Av_i}{|v_i|^2} = \lambda_i$.

Expander Graph:

All $x \perp \bar{\mathbf{1}}$: $\sum_i x_i = 0$.

Degree d, think of random ± 1 labels.

$(Ax)_i =$ average value of labels.

Roughly $\pm\sqrt{d}$.

Expander: "no good cuts" "all cuts same".

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

$\frac{x^T A x}{d|x|^2}$ is proxy for "eigenvalue".

$x^T(dI - A)x = d|x|^2 - x^T A x$.

If different for different x, then λ_1/λ_n is large:

$Av_i = \lambda_i v_i \rightarrow \frac{v_i^T A v_i}{|v_i|^2} = \lambda_i$.

Expander Graph:

All $x \perp \bar{\mathbf{1}}$: $\sum_i x_i = 0$.

Degree d, think of random ± 1 labels.

$(Ax)_i =$ average value of labels.

Roughly $\pm\sqrt{d}$.

Expander: "no good cuts" "all cuts same".

$\implies$ every ± 1 vector is about the same!

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{1}$ has eigenvalue 0.

$\frac{x^T Ax}{d|x|^2}$ is proxy for "eigenvalue".

$x^T(dI - A)x = d|x|^2 - x^T Ax$.

If different for different x, then λ_1/λ_n is large:

$Av_i = \lambda_i v_i \to \frac{v_i^T Av_i}{|v_i|^2} = \lambda_i$.

Expander Graph:

All $x \perp \bar{1}$: $\sum_i x_i = 0$.

Degree d, think of random ± 1 labels.

$(Ax)_i =$ average value of labels.

Roughly $\pm\sqrt{d}$.

Expander: "no good cuts" "all cuts same".

$\implies$ every ± 1 vector is about the same!

Fast convergence of iterative method.

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

$\frac{x^TAx}{d|x|^2}$ is proxy for “eigenvalue”.

$x^T(dI - A)x = d|x|^2 - x^TAx$.

If different for different x, then λ_1/λ_n is large:

$Av_i = \lambda_i v_i \to \frac{v_i^TAv_i}{|v_i|^2} = \lambda_i$.

Expander Graph:

All $x \perp \bar{\mathbf{1}}$: $\sum_i x_i = 0$.

Degree d, think of random ± 1 labels.

$(Ax)_i =$ average value of labels.

Roughly $\pm\sqrt{d}$.

Expander: “no good cuts” “all cuts same”.

$\implies$ every ± 1 vector is about the same!

Fast convergence of iterative method.

Intuitively: fast communication across graph,

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

$\frac{x^T Ax}{d|x|^2}$ is proxy for "eigenvalue".

$x^T(dI - A)x = d|x|^2 - x^T Ax$.

If different for different x, then λ_1/λ_n is large:

$Av_i = \lambda_i v_i \rightarrow \frac{v_i^T A v_i}{|v_i|^2} = \lambda_i$.

Expander Graph:

All $x \perp \bar{\mathbf{1}}$: $\sum_i x_i = 0$.

Degree d, think of random ± 1 labels.

$(Ax)_i =$ average value of labels.

Roughly $\pm\sqrt{d}$.

Expander: "no good cuts" "all cuts same".

$\implies$ every ± 1 vector is about the same!

Fast convergence of iterative method.

Intuitively: fast communication across graph,

$\implies$ fast iterative algorithm.

Examples: Eigenvectors.

For graph laplacians: $L = dI - A$.

$dI - A$. Constant eigenvector, $\bar{\mathbf{1}}$ has eigenvalue 0.

$\frac{x^T Ax}{d|x|^2}$ is proxy for "eigenvalue".

$x^T(dI - A)x = d|x|^2 - x^T Ax$.

If different for different x, then λ_1/λ_n is large:

$Av_i = \lambda_i v_i \to \frac{v_i^T Av_i}{|v_i|^2} = \lambda_i$.

Expander Graph:

All $x \perp \bar{\mathbf{1}}$: $\sum_i x_i = 0$.

Degree d, think of random ± 1 labels.

$(Ax)_i =$ average value of labels.

Roughly $\pm\sqrt{d}$.

Expander: "no good cuts" "all cuts same".

$\implies$ every ± 1 vector is about the same!

Fast convergence of iterative method.

Intuitively: fast communication across graph,

$\implies$ fast iterative algorithm.

Path (or Cycle)

Consider $\frac{x^T A x}{d|x|^2}$.

Path (or Cycle)

Consider $\frac{x^T Ax}{d|x|^2}$.

Random ± 1, same argument as for expander.

Path (or Cycle)

Consider $\frac{x^T A x}{d|x|^2}$.

Random ± 1, same argument as for expander.

$+1/-1$ edges change mass a lot.

Path (or Cycle)

Consider $\frac{x^T Ax}{d|x|^2}$.

Random ± 1, same argument as for expander.

$+1/-1$ edges change mass a lot.

$\implies \frac{x^T Ax}{d|x|^2} << 1$.

Path (or Cycle)

Consider $\frac{x^T Ax}{d|x|^2}$.

Random ± 1, same argument as for expander.

$+1/-1$ edges change mass a lot.

$\implies \frac{x^T Ax}{d|x|^2} << 1$.

Good cut: $+1$ on left half, -1 on right half.

Path (or Cycle)

Consider $\frac{x^T Ax}{d|x|^2}$.

Random ± 1, same argument as for expander.

$+1/-1$ edges change mass a lot.

$\implies \frac{x^T Ax}{d|x|^2} << 1$.

Good cut: $+1$ on left half, -1 on right half.

Lose mass only in middle.

Path (or Cycle)

Consider $\frac{x^T A x}{d|x|^2}$.

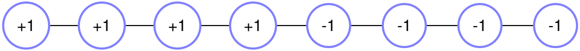
Random ± 1, same argument as for expander.

$+1/-1$ edges change mass a lot.

$\implies \frac{x^T A x}{d|x|^2} << 1$.

Good cut: $+1$ on left half, -1 on right half.

Lose mass only in middle.



$\implies \frac{x^T A x}{d|x|^2} \approx 1 - \frac{1}{n}$.

Path (or Cycle)

Consider $\frac{x^T Ax}{d|x|^2}$.

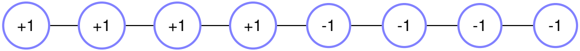
Random ± 1, same argument as for expander.

$+1/-1$ edges change mass a lot.

$\implies \frac{x^T Ax}{d|x|^2} << 1$.

Good cut: $+1$ on left half, -1 on right half.

Lose mass only in middle.



$\implies \frac{x^T Ax}{d|x|^2} \approx 1 - \frac{1}{n}$.

$\implies \lambda_1$ and λ_n differ by a factor of n.

Path (or Cycle)

Consider $\frac{x^T Ax}{d|x|^2}$.

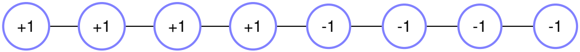
Random ± 1, same argument as for expander.

$+1/-1$ edges change mass a lot.

$\implies \frac{x^T Ax}{d|x|^2} << 1$.

Good cut: $+1$ on left half, -1 on right half.

Lose mass only in middle.



$\implies \frac{x^T Ax}{d|x|^2} \approx 1 - \frac{1}{n}$.

$\implies \lambda_1$ and λ_n differ by a factor of n.

Worse example:

Path (or Cycle)

Consider $\frac{x^T Ax}{d|x|^2}$.

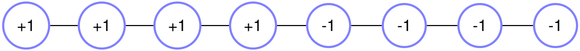
Random ± 1, same argument as for expander.

$+1/-1$ edges change mass a lot.

$\implies \frac{x^T Ax}{d|x|^2} << 1$.

Good cut: $+1$ on left half, -1 on right half.

Lose mass only in middle.



$\implies \frac{x^T Ax}{d|x|^2} \approx 1 - \frac{1}{n}$.

$\implies \lambda_1$ and λ_n differ by a factor of n.

Worse example:

Path (or Cycle)

Consider $\frac{x^T A x}{d|x|^2}$.

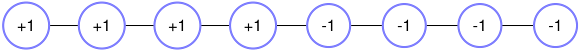
Random ± 1, same argument as for expander.

$+1/-1$ edges change mass a lot.

$\implies \frac{x^T A x}{d|x|^2} << 1$.

Good cut: $+1$ on left half, -1 on right half.

Lose mass only in middle.



$\implies \frac{x^T A x}{d|x|^2} \approx 1 - \frac{1}{n}$.

$\implies \lambda_1$ and λ_n differ by a factor of n.

Worse example:

$\implies \frac{x^T A x}{|x|^2} \approx 1 - \frac{1}{n^2}$.

Path (or Cycle)

Consider $\frac{x^T Ax}{d|x|^2}$.

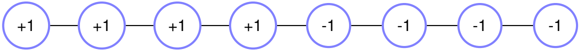
Random ± 1, same argument as for expander.

$+1/-1$ edges change mass a lot.

$\implies \frac{x^T Ax}{d|x|^2} << 1$.

Good cut: $+1$ on left half, -1 on right half.

Lose mass only in middle.



$\implies \frac{x^T Ax}{d|x|^2} \approx 1 - \frac{1}{n}$.

$\implies \lambda_1$ and λ_n differ by a factor of n.

Worse example:

$\implies \frac{x^T Ax}{|x|^2} \approx 1 - \frac{1}{n^2}$. For Laplacian $1/n^2$.

Path (or Cycle)

Consider $\frac{x^T Ax}{d|x|^2}$.

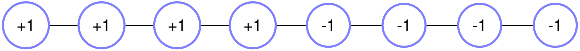
Random ± 1, same argument as for expander.

$+1/-1$ edges change mass a lot.

$\implies \frac{x^T Ax}{d|x|^2} << 1$.

Good cut: $+1$ on left half, -1 on right half.

Lose mass only in middle.



$\implies \frac{x^T Ax}{d|x|^2} \approx 1 - \frac{1}{n}$.

$\implies \lambda_1$ and λ_n differ by a factor of n.

Worse example:

$\implies \frac{x^T Ax}{|x|^2} \approx 1 - \frac{1}{n^2}$. For Laplacian $1/n^2$.

Convergence is $O(n^2)$.

Path (or Cycle)

Consider $\frac{x^TAx}{d|x|^2}$.

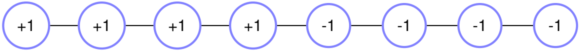
Random ± 1, same argument as for expander.

$+1/-1$ edges change mass a lot.

$\implies \frac{x^TAx}{d|x|^2} << 1$.

Good cut: $+1$ on left half, -1 on right half.

Lose mass only in middle.



$\implies \frac{x^TAx}{d|x|^2} \approx 1 - \frac{1}{n}$.

$\implies \lambda_1$ and λ_n differ by a factor of n.

Worse example:

$\implies \frac{x^TAx}{|x|^2} \approx 1 - \frac{1}{n^2}$. For Laplacian $1/n^2$.

Convergence is $O(n^2)$. Or $O(n)$.