Quick Tour of Linear Algebra and Graph Theory

CS224w: Social and Information Network Analysis
Fall 2012 Yu "Wayne" Wu
Based on Borja Pelato's version in Fall 2011

Matrices and Vectors

- Matrix: A rectangular array of numbers, e.g., $A \in \mathbb{R}^{m \times n}$:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

- Vector: A matrix consisting of only one column (default) or one row, e.g., $x \in \mathbb{R}^n$

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

Matrix Multiplication

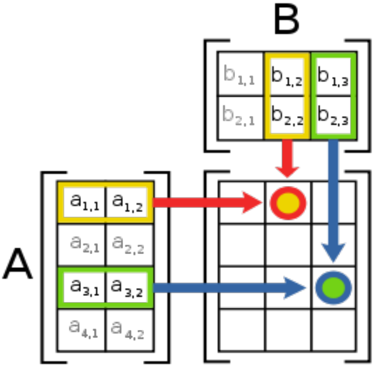
- If $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{n \times p}$, $C = AB$, then $C \in \mathbb{R}^{m \times p}$:

$$C_{ij} = \sum_{k=1}^{n} A_{ik} B_{kj}$$

- Special cases: Matrix-vector product, inner product of two vectors. e.g., with $x, y \in \mathbb{R}^n$:

$$x^T y = \sum_{i=1}^{n} x_i y_i \in \mathbb{R}$$

Matrix Multiplication



Properties of Matrix Multiplication

- Associative: $(AB)C = A(BC)$
- Distributive: $A(B + C) = AB + AC$
- Non-commutative: $AB \neq BA$

Proof of associativity:
Let $L = (AB)C$ and $R = A(BC)$, then we can show
$L_{ij} = \sum_{k=1}^{p} \sum_{l=1}^{n} (a_{il} \circ b_{lk}) \circ c_{kj} = \sum_{l=1}^{n} \sum_{k=1}^{p} a_{il} \circ (b_{lk} \circ c_{kj}) = R_{ij}$.

Operators and properties

- Transpose: $A \in \mathbb{R}^{m \times n}$, then $A^T \in \mathbb{R}^{n \times m}$: $(A^T)_{ij} = A_{ji}$
- Properties:
  - $(A^T)^T = A$
  - $(AB)^T = B^T A^T$
  - $(A + B)^T = A^T + B^T$

Identity Matrix

- Identity matrix: $I = I_n \in \mathbb{R}^{n \times n}$:

$$I_{ij} = \begin{cases} 1 & \text{i=j,} \\ 0 & \text{otherwise.} \end{cases}$$

- $\forall A \in \mathbb{R}^{m \times n}$: $AI_n = I_mA = A$

$$I_1 = \begin{bmatrix} 1 \end{bmatrix}, \ I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \ I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \ \cdots, \ I_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

Diagonal Matrix

- Diagonal matrix: $D = diag(d_1, d_2, \ldots, d_n)$:

$$D_{ij} = \begin{cases} d_i & \text{j=i,} \\ 0 & \text{otherwise.} \end{cases}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & -3 \end{bmatrix}$$

Other Special Mtrices

- Symmetric matrices: $A \in \mathbb{R}^{n \times n}$ is symmetric if $A = A^T$.
- Orthogonal matrices: $U \in \mathbb{R}^{n \times n}$ is orthogonal if $UU^T = I = U^T U$

Linear Independence and Rank

- A set of vectors $\{x_1, \ldots, x_n\}$ is linearly independent if $\nexists \{\alpha_1, \ldots, \alpha_n\}$: $\sum_{i=1}^{n} \alpha_i x_i = 0$
- Rank: $A \in \mathbb{R}^{m \times n}$, then $rank(A)$ is the maximum number of linearly independent columns (or equivalently, rows)
- Properties:
  - $rank(A) \leq \min\{m, n\}$
  - $rank(A) = rank(A^T)$
  - $rank(AB) \leq \min\{rank(A), rank(B)\}$
  - $rank(A + B) \leq rank(A) + rank(B)$

Rank from row-echelon forms

$$\begin{bmatrix} 1 & 2 & 1 \\ -2 & -3 & 1 \\ 3 & 5 & 0 \end{bmatrix} R_2 \rightarrow 2r_1 + r_2 \begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 3 \\ 3 & 5 & 0 \end{bmatrix} R_3 \rightarrow -3r_1 + r_3 \begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 3 \\ 0 & -1 & -3 \end{bmatrix}$$

$$R_3 \rightarrow r_2 + r_3 \begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & 0 \end{bmatrix} R_1 \rightarrow -2r_2 + r_1 \begin{bmatrix} 1 & 0 & -5 \\ 0 & 1 & 3 \\ 0 & 0 & 0 \end{bmatrix}$$

Matrix Inversion

- If $A \in \mathbb{R}^{n \times n}$, $rank(A) = n$, then the inverse of A, denoted A^{-1} is the matrix that: $AA^{-1} = A^{-1}A = I$
- Properties:
  - $(A^{-1})^{-1} = A$
  - $(AB)^{-1} = B^{-1}A^{-1}$
  - $(A^{-1})^T = (A^T)^{-1}$
- The inverse of an orthogonal matrix is its transpose

Eigenvalues and Eigenvectors

- $A \in \mathbb{R}^{n \times n}$, $\lambda \in \mathbb{C}$ is an eigenvalue of A with the corresponding eigenvector $x \in \mathbb{C}^n$ ($x \neq 0$) if:

$$Ax = \lambda x$$

- eigenvalues: the n possibly complex roots of the polynomial equation $det(A - \lambda I) = 0$, and denoted as $\lambda_1, \ldots, \lambda_n$

$$A\mathbf{x} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 3 \\ -3 \end{bmatrix} = \begin{bmatrix} 2 \cdot 3 + 1 \cdot (-3) \\ 1 \cdot 3 + 2 \cdot (-3) \end{bmatrix} = \begin{bmatrix} 3 \\ -3 \end{bmatrix} = 1 \cdot \begin{bmatrix} 3 \\ -3 \end{bmatrix}.$$

Eigenvalues and Eigenvectors Properties

- Usually eigenvectors are normalized to unit length.
- If A is symmetric, then all the eigenvalues are real and the eigenvectors are orthogonal to each other.
- $tr(A) = \sum_{i=1}^{n} \lambda_i$
- $det(A) = \prod_{i=1}^{n} \lambda_i$
- $rank(A) = |\{1 \leq i \leq n | \lambda_i \neq 0\}|$

Matrix Eigendecomposition

$A \in \mathbb{R}^{n \times n}$, $\lambda_1, \ldots, \lambda_n$ the eigenvalues, and $x_1, \ldots, x_n$ the eigenvectors. $P = [x_1 | x_2 | \ldots | x_n]$, $D = diag(\lambda_1, \ldots, \lambda_n)$, then:

$$
\begin{aligned}
\mathsf{A}\,\mathsf{P} &= \mathsf{A}\begin{bmatrix}\mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_k\end{bmatrix} \\
&= \begin{bmatrix}\mathsf{A}\mathbf{X}_1 & \mathsf{A}\mathbf{X}_2 & \cdots & \mathsf{A}\mathbf{X}_k\end{bmatrix} \\
&= \begin{bmatrix}\lambda_1\mathbf{X}_1 & \lambda_2\mathbf{X}_2 & \cdots & \lambda_k\mathbf{X}_k\end{bmatrix} \\
&= \begin{bmatrix}\lambda_1 x_{11} & \lambda_2 x_{21} & \cdots & \lambda_k x_{k1} \\ \lambda_1 x_{12} & \lambda_2 x_{22} & \cdots & \lambda_k x_{k2} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_1 x_{1k} & \lambda_2 x_{2k} & \cdots & \lambda_k x_{kk}\end{bmatrix} \\
&= \begin{bmatrix}x_{11} & x_{21} & \cdots & x_{k1} \\ x_{12} & x_{22} & \cdots & x_{k2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1k} & x_{2k} & \cdots & x_{kk}\end{bmatrix}\begin{bmatrix}\lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_k\end{bmatrix} \\
&= \mathsf{P}\,\mathsf{D},
\end{aligned}
$$

Matrix Eigendecomposition

- Therefore, $A = PDP^{-1}$.
- In addition:

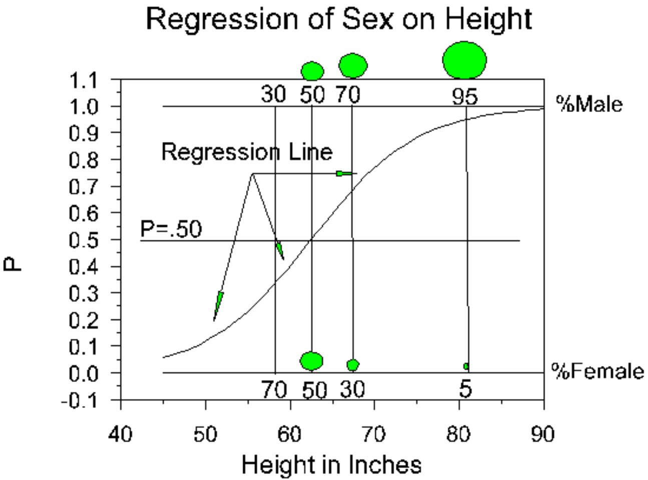
  $$A^2 = (PDP^{-1})(PDP^{-1}) = PD(P^{-1}P)DP^{-1} = PD^2P^{-1}$$

  By induction, $A^n = PD^nP^{-1}$.
- A special case of Singular Value Decomposition

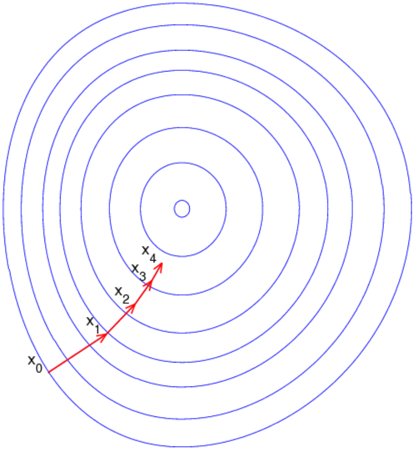
Convex Optimization

- A set of points S is convex if, for any $x, y \in S$ and for any $0 \leq \theta \leq 1$,
$$\theta x + (1 - \theta) y \in S$$
- A function $f : S \to \mathbb{R}$ is convex if its domain S is a convex set and
$$f(\theta x + (1 - \theta) y) \leq \theta f(x) + (1 - \theta) f(y)$$
for all $x, y \in S$, $0 \leq \theta \leq 1$.

Logistic Regression



Gradient Descent



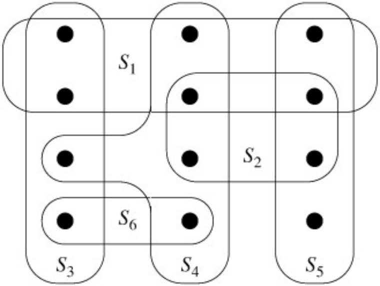
Submodularity

- A function $f : S \to \mathbb{R}$ is submodular if for any subset $A \subseteq B$,

$$f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B)$$

- Submodular functions allow approximate discrete optimization.

Greedy Set Cover



If the optimal solution contains m sets, greedy algorithm finds a set cover with at most $m \log_e n$ sets.

Proofs

Induction:

1. Show result on base case, associated with $n = k_0$
2. Assume result true for $n \leq i$. Prove result for $n = i + 1$
3. Conclude result true for all $n \geq k_0$

Example:
For all natural number n, $1 + 2 + 3 + ... + n = \frac{n*(n+1)}{2}$
Base case: when $n = 1$, $1 = 1$.
Assume statement holds for $n = k$, then
$1 + 2 + 3 + ... + k = \frac{k*(k+1)}{2}$.
We see $1 + 2 + 3 + ... + (k + 1) = \frac{k*(k+1)}{2} + (k + 1) = \frac{(k+1)(k+2)}{2}$.

Proofs

Contradiction (reductio ad absurdum):

1. Assume result is false
2. Follow implications in a deductive manner, until a contradiction is reached
3. Conclude initial assumption was wrong, hence result true

Example:
Let's try to prove there is no greatest even integer.
First suppose there is one, name N, and for any even integer n, we have $N \geq n$.
Now let define M as $N + 2$. Then we see M is even, but also greater than N. Thus by contradiction we prove the statement.

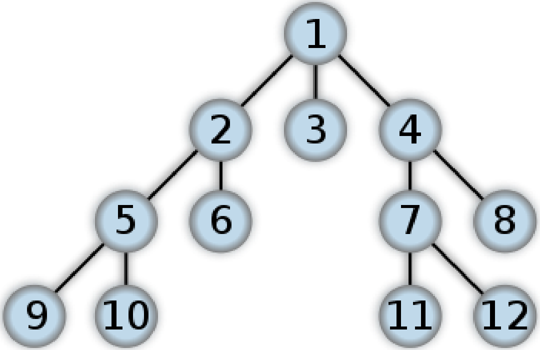
Graph theory

- Definitions: vertex/node, edge/link, loop/cycle, degree, path, neighbor, tree, clique,...
- Random graph (Erdos-Renyi): Each possible edge is present with some probability p
- (Strongly) connected component: subset of nodes that can all reach each other
- Diameter: longest minimum distance between two nodes
- Bridge: edge connecting two otherwise disjoint connected components

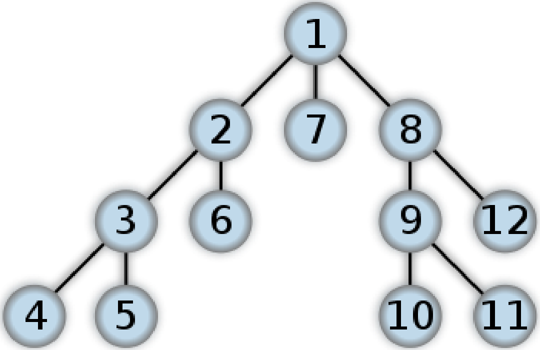
Basic algorithms

- BFS: explore by “layers”
- DFS: go as far as possible, then backtrack
- Greedy: maximize goal at each step
- Binary search: on ordered set, discard half of the elements at each step

BFS



DFS



Complexity

- Number of operations as a function of the problem parameters.
- Examples
  1. Find shortest path between two nodes:
     - DFS: very bad idea, could end up with the whole graph as a single path
     - BFS from origin: good idea
     - BFS from origin and destination: even better!
  2. Given a node, find its connected component
     - Loop over nodes: bad idea, needs N path searches
     - BFS or DFS: good idea