**Linear algebra basics**

Rank r = number of rows which are linear independent.

M is symmetric -> M = MT.

∥N∥ = Frobenius norm = the square root of the sum of the squares of the elements of N.

Eigen-decomposition

Me = λe (λ = eigenvalue, e = eigenvector)

[e: unit length, first nonzero value positive]

* det(M – λI) = 0

M = X**ΛXT**

X = eigenvector matrix; 

**Dimensionality Reduction**

Dimensionality reduction

* compress/reduce dimension
  + reason: 1. Discover hidden correlations. 2. Remove redundant and noisy. 3. Interpretation and visualization. 4. Easy storage and processing
* reconstruct the original matrix by two or more smaller matrices

Singular value decomposition (SVD)

* A=𝑈Σ𝑉T= Σi𝜎i𝑢i∘𝑣iT
* **𝑈** (m x r, 𝑢i =left-signular vectors of A), each cols are eigenvectors of AA\*,  
  **𝑉** (n x r, 𝑣I = right-signular vectors of A): each cols are eigenvectors of A\*A  
  U and V are column orthonormal matrix -> UUT=UTU=I  
  **Σ** (r x r, 𝜎i = non-negative singular values of A):diagonal, square roots of eigenvalues of both AA\* and A\*A, sort in decreasing order
* **Low rank approximation (优点)**

1. Set smaller singular value to 0
2. Eliminate correspond row of U and V

𝑆= diagonal 𝑘×𝑘 matrix where 𝑠i=𝜎i (𝑖=1…𝑘) and 𝑠i=0(𝑖>𝑘)

* Best rank-k approximation



* + Reason: ui and vi are **unit length**, 𝜎i scales the value, smallest introduces less error.
  + Pratices: keep 80% to 90% of energy 
* **SVD Draw back** (Interpretability problem & lack of sparsity)
* **Query using concepts (=qV)**

Project query to concept space.

Measure similarity by cosine distance.

CUR decomposition

decompose a sparse matrix into spares, smaller matrices

- 𝑋≈𝐶𝑈𝑅

𝐶:set of columns of 𝑋 in random. 𝑅: set of rows of 𝑋 in random. U: pseudo-inverse of the intersection of the chosen rows and columns.

More important rows/cols are more likely to be picked.

Middle matrix

Pros: easy interpretation (C, R are real values)  
Cons: duplicate cols and rows (large norm)

Principle component analysis (PCA)

* Application of SVD
* reconstruct data matrix by a smaller number of eigenvectors
* maximize variance, minimize root-mean-square error

**Recommender System**

can be classified into two groups

1. Content-based filtering: examine properties of the items recommended, (e.g. recommend same movie which has same genre), can recommend new item.
2. Collaborative filtering: recommend items based on similarity measures between users and/or items. Cannot recommend new item.

Matrix Factorization  
Low rank matrix factorization framework.

**Fully observed matrix**

1. LU Decomposition, A = LU  
   U: use Gaussian elimination  
   L: apply inverse operation on the corresponding entry to I to get L
2. SVD (上一章节有).

**Partially Observed Matrix**

1. Probabilistic Matrix Factorization (PMF)  
   factorize into product of two low-rank matrices.  
     
     
     
   PMF objective function: (MAP=minimize below function)  
     
     
   Iij indicator, if user i rated item j, then = 1, else = 0.  
   1st item is sum-of-squared-error  
   2nd and 3rd item are quadratic regularization term.
2. Non-negative matrix factorization (NMF)

Factorize a matrix into the product of two non-negative matrices, can be used to learn the "parts"   
 



**Massive Link Analysis**

**#Web as a graph**  
nodes: webpages; edges: hyperlinks;  
**#PageRank**  
idea: in-links counts as votes, in-links from important page counts more.

Matrix M, where  if i -> j (sum of col = 1). di = number of out-links at i.

Rank vector r, where .

* r = Mr (r is the principal eigenvector with eigenvalue = 1)

Power iteration method:  
obj: to find the principal eigenvector.  
1. Initialize: 

2. Iteration:   
3. Stop at it convergence: 

Stationary distribution  
random walk reach a state where  


**PageRank problems:**

Spider traps (all out-links within the group)  
dead ends (no out-links)  
solution: teleport

Reason why teleport works: **Markov chains**

For any start vector, the power method applied to a Markov transition matrix **P** will converge to a unique positive stationary vector as long as **P** is **stochastic**, **irreducible** and **aperiodic**.

**PageRank equations**:  
  


->  (too large for storage)

n = number of nodes; , e is vector of all 1’s.

improvement:  
-  (no dead ends)

- Sparse matrix encoding (store only the links, fit disk)

- block-based… (break rnew into k blocks)

- block-strip… (break M into stripes, each stripe only contains destination in block rnew)

**#Topic-specific page rank**

  
|S| is the size of set S (teleport set, topic-specific set)