Matrix Decompositions

Tuesday, 18 February, 2020 2:50 PM

Source: https://people.duke.edu/~ccc14/sta-663/LinearAlgebraMatrixDecompWithSolutions.html

Matrix Decompositions are an important step in solving linear systems in a computationally efficient manner.

LU Decompositions and Gaussian Elimination

LU = 'Lower Upper'.

1) LU decompositions of matrix A: A = LU , where L is lower triangular and U is upper triangular.
2) LU decompositions basically gaussian elimination, but only work with matrix A rather than augmented matrix.
3) Goal: factorization of a square matrix into two triangular matrices; L and U, such that product of L and U = A.

Source: https://www.geeksforgeeks.org/l-u-decomposition-system-linear-equations/

Steps for L U Decomposition:

Given a set of linear equations, convert into matrix form A X = C where A = coefficient, X = variable matrix, C= matrix of numbers on the right-hand side of the equations.

Reduce coefficient matrix A to row echelon form using Gaussian elimination. Matrix obtained is U.

To find L, two methods:

- Assume the remaining elements as some artificial variables, make equations using A = L U and solve them to find those artificial variables
- Remaining elements are the multiplier coefficients because of which the respective positions became zero in the U matrix

Additional info:

Now have A (the nXn coefficient matrix), L (the nXn lower triangular matrix), U (the nXn upper triangular matrix), X (the nX1 matrix of variables) and C (the nX1 matrix of numbers on the right-hand side of the equations).

The given system of equations is A X = C. We substitute A = L U. Thus, we have L U X = C.

We put Z = U X, where Z is a matrix or artificial variables and solve for L Z = C first and then solve for U X = Z to find X or the values of the variables, which was required.

Recommendation system links:

https://heartbeat.fritz.ai/recommender-systems-with-python-part-iii-collaborative-filtering-singular-value-decomposition-5b5dcb3f242b

https://beckernick.github.io/matrix-factorization-recommender/

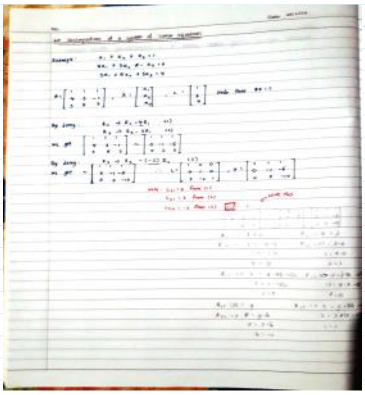
https://www.kaggle.com/gohngeejuay/recommender-systems-in-python-101/edit

Math: https://www.geeksforgeeks.org/l-u-decomposition-system-linear-equations/

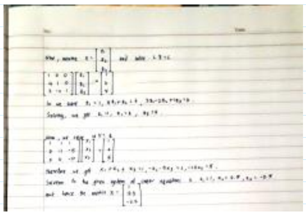
https://people.duke.edu/~ccc14/sta-663/LinearAlgebraMatrixDecompWithSolutions.html

Notes to make: Matrix Decompositions

- Matrix Decompositions
- Cosine similarity



Additional info: Solving systems of linear equations after using L U Decompositions



Reasons for matrix decompositions in movie recommendation:

Matrix factorization can be used to discover features underlying the interactions between two different kinds of entities

From <https://heartbeat.fritz.ai/recommender-systems-with-python-part-iii-collaborative-filtering-singular-value-decomposition-5b5dcb3f242b>

- One advantage of employing matrix factorization for recommender systems is the fact that it can incorporate implicit feedback—information that's not directly given but can be derived by analysing user behaviour—such as items frequently bought or viewed
- If we can discover these kinds of latent features (like genre or actors and directors), we should be able to predict a rating with respect to a certain user and a certain item, because the features associated with the user should match with the features associated with the item.

QR Decomposition

Source: http://mathworld.wolfram.com/QRDecomposition.html

Other references: https://genomicsclass.github.io/book/pages/qr_and_regression.html

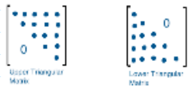
Given a matrix A, its QR decomposition is a matrix decomposition of the form:

A = Q R

Where R is an upper triangular matrix and Q is an orthogonal matrix, ie one satisfying

$Q^T Q = I$ where Q^T is the transpose of Q and I is the identitiy matrix.

This matrix can be used to solve linear systems of equations.



Cholesky Decomposition

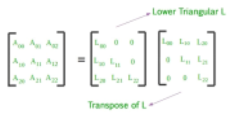
Source: https://www.geeksforgeeks.org/cholesky-decomposition-matrix-decomposition/

Given a symmetric positive defined matrix A, the Cholesky decomposition is an upper triangular matrix U with strictly positive diagonal entries such that:

$A = U^T U$

The Cholesky decomposition is roughly twice as efficient as the LU decomposition for solving systems of linear equations.

From <https://www.geeksforgeeks.org/cholesky-decomposition-matrix-decomposition/>



Singular Value Decomposition

Source: https://blog.statsbot.co/singular-value-decompos

https://www.quora.com/What-is-an-intuitive-explanation

To find a SVD of A, we must find V, \Sigma and U such tha

A=UΣV^T

1. V must diagonalize A^TA

1.1. V_i are eigenvectors of A^TA

2. Σ where Σ_ii are singular values of A.

3. U must diagonalize AA^T

3.1 u_i are eigenvectors of AA^T.

If A has rank r then:

1. vi⋯vr forms an orthonormal basis for the range of A^T

2. ui⋯ur form an orthonormal basis for the range of A

3. Rank of A is equal to the number of nonzero entries of

We see that we can express A another way, it can be show matrixes.

A=∑ri=1σiuivTi

From <https://www.quora.com/What-is-an-intuitive-explanation-of-si

We know that by construction σi monotonic decreasing, t decreases. This means that the summation to k<r is an ap

sition-tutorial-52c695315254
n-of-singular-value-decomposition-SVD

at:

S. From the form of this factorization

wn that A can be written as a sum of Rank = 1

ngular-value-decomposition-SVD>

the significance/weight of the nth term
proximation A^ of rank k for the matrix A.

Cosine Similarity

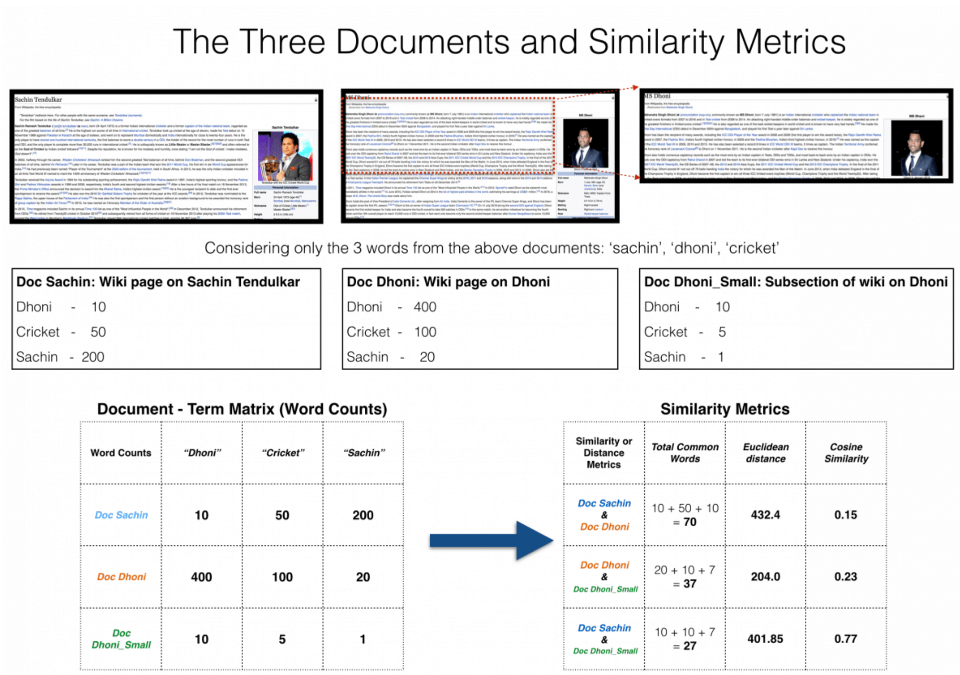
Friday, 21 February, 2020 10:38 AM

Cosine similarity is a metric used to measure how similar the documents are irrespective of their size. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. The cosine similarity is advantageous because even if the two similar documents are far apart by the Euclidean distance (due to the size of the document), chances are they may still be oriented closer together. The smaller the angle, higher the cosine similarity.

From <https://www.machinelearningplus.com/nlp/cosine-similarity/>

Formula:

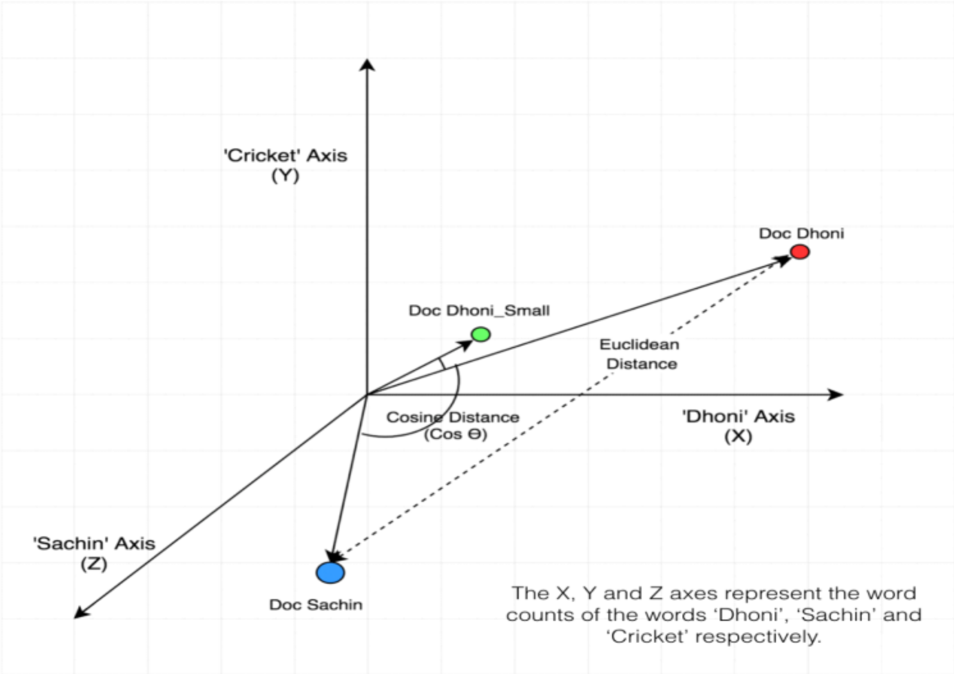
$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}},$$



Word Counts	"Dhoni"	"Cricket"	"Sachin"
Doc Sachin	10	50	200
Doc Dhoni	400	100	20
Doc Dhoni_Small	10	5	1

Similarity or Distance Metrics	Total Common Words	Euclidean distance	Cosine Similarity
Doc Sachin & Doc Dhoni	10 + 50 + 10 = 70	432.4	0.15
Doc Dhoni & Doc Dhoni_Small	20 + 10 + 7 = 37	204.0	0.23
Doc Sachin & Doc Dhoni_Small	10 + 10 + 7 = 27	401.85	0.77

Projection of Documents in 3D Space



2`