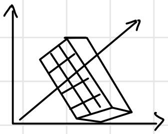
2/22/22

matrix and linear algebraic property:

1)

2) Dimensionality Reduction (calc 3)



Linear Algebra Review:

- linearly independent: $a_1\vec{v}_1 + \ldots + a_n\vec{v}_n = \vec{0}$ $\boxed{\text{for } a_i = 0}$
- Determinant: $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ $\det(A) = ad - bc$

  Also works for 3x3 (can define it recursively)
- create matrix for n dimensional space
- $m > n$? Sol: No
- rank : dimension of vector space

  for $n \times n$ dataset, rank is the full column list

Approximation

1) Use rank to reduce values

   For $n \times m$ matrix, storing takes $m \cdot n$ values

   Use Rank. rank of matrix is k. $\therefore$ storing is $k(n+m)$ values

   $\because k < m$

   $\therefore$ using rank reduces ✓

2) Forbieus

3) rank-k approximation: $\underbrace{A^{(k)}}_{\substack{m\times n \text{ matrix} \\ \text{w/ rank } k}} = \underset{\{B \mid \text{rank}(B)=k\}}{\arg\min} \underbrace{d_F(A, B)}_{\substack{\text{mininze the} \\ \text{Forbieus function}}}$

   Goal: find matrix B

   approximate matrix w/ lower Rank matrix to define A.

4) SVD: singular Value Decomposition. $A = U \Sigma V^T$

   - U, V are othogonal & unit length

   $\Sigma = \begin{bmatrix} \sigma_1 & & & 0 \\ & \sigma_2 & & \\ & & \ddots & \\ 0 & & & \sigma_r \end{bmatrix}$ ← Diagonal matrix

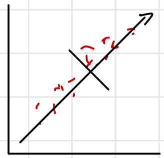
   - $A^{(k)} = U_1 \Sigma_1 V_1^T$ has rank k.

ex: $\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$ Rank 2 ∵ 2 unique col

Approximation gives: $\begin{matrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$

- property: the larger the k, the smaller the distance
- the i^{th} singular vector is the direction of the i^{th} most variance

i.e.

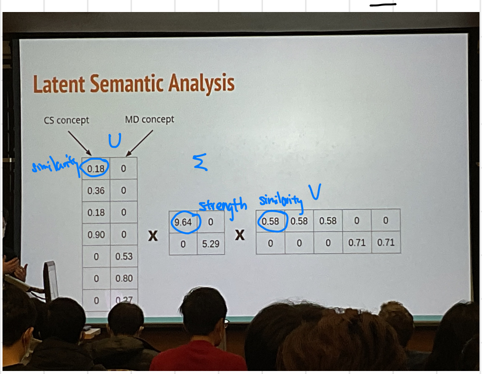


this is the direction
(ith column of U)

Principle Component Analysis

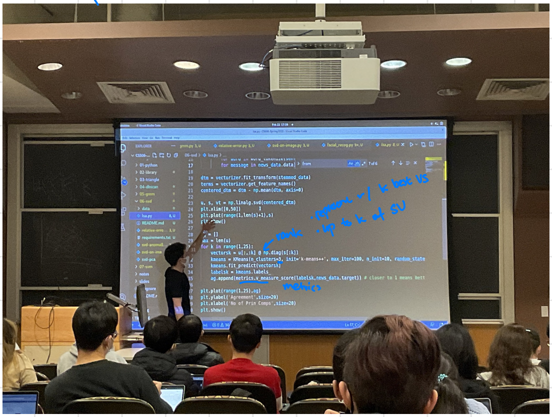
Latent Semantic Analysis

take matrix and break into 3 matrices (U, V, Σ)



- better represent each doc by: frequency ($^n/\Sigma n$) and TfiDf

HW Help:


for message in news_data.data]
dtm = vectorizer.fit_transform(stemmed_data)
terms = vectorizer.get_feature_names()
centered_dtm = dtm - np.mean(dtm, axis=0)
u, s, vt = np.linalg.svd(centered_dtm)
plt.xlim([0,50])
plt.plot(range(1,len(s)+1),s)
for k in range(1,25):
vectorsk = u[:,:k] @ np.diag(s[:k])
kmeans = KMeans(n_clusters=3, init='k-means++', max_iter=100, n_init=10, random_state
kmeans.fit_predict(vectorsk)
labelsk = kmeans.labels_
ag.append(metrics.v_measure_score(labelsk,news_data.target)) # closer to 1 means bett
plt.plot(range(1,25),ag)
plt.ylabel('Agreement',size=20)
plt.xlabel('No of Prin Comps',size=20)
plt.show()
· represent w/ k best VS
· up to k of SV
rank
metrics