Data Mining and Machine Learning

Madhavan Mukund

Lecture 23, Jan–Apr 2020

https://www.cmi.ac.in/~madhavan/courses/dmml2020jan/

Queries and responses

Two classic problems in natural languages

Synonymy Different words for the same concept

- {car, automobile}, {picture, image, photo}

Polysemy Words have multiple meanings

- Jaguar the car, vs jaguar the animal

Vector space representation does not tackle these problems

- Recall, cosine similarity between query q and document d, $q \cdot d$
- Synonymy leads to underestimating $q \cdot d$ — q and d use different words for same concept
- Polysemy leads to overestimating $q \cdot d$ — same word has different interpretation in q and d

A concept space

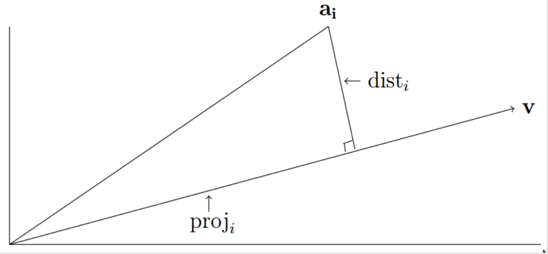
- Ideally, the building blocks of documents are concepts
  - Different words may map to the same concept — synonymy
  - The same word may map to multiple concepts — polysemy
- Transform document representation from vector over terms to vector over concepts
  - In the language of linear algebra, find an alternate basis for document space
- Quantify correlation between words and concepts

Singular Value Decomposition (SVD)

- Term-document matrix M, dimensions $n \times d$
  - Rows are terms, columns are documents
  - $M[i,j]$ is TF-IDF score for term i in document j
- Decompose M as $UDV^\top$
  - D is a $k \times k$ diagonal matrix, positive real entries
  - U is $n \times k$, V is $d \times k$
  - Columns of U, V are orthonormal — unit vectors, mutually orthogonal
- Interpretation
  - Columns of V correspond to new abstract concepts
  - Rows of U describe decomposition of terms across concepts
  - For columns $\mathbf{u}_i$ of U and $\mathbf{v}_i$ of V, $\mathbf{u}_i \cdot \mathbf{v}_i^\top$ is an $n \times d$ matrix, like M
  - $\mathbf{u}_i \cdot \mathbf{v}_i^\top$ describes components of rows of M along direction $\mathbf{v}_i$

Singular vectors

- Unit vectors passing through the origin
- Want to find "best" k singular vectors to represent concept space
- Suppose we project $\mathbf{a}_i = (a_{i1}, a_{i2}, \ldots, a_{id})$ onto $\mathbf{v}$ through origin



- Minimizing distance of $\mathbf{a}_i$ from v is equivalent to maximizing the projection of $\mathbf{a}_i$ onto v
- Length of the projection is $\mathbf{a}_i \cdot \mathbf{v}$

Singular vectors ...

- Sum of squares of lengths of projections of all rows in M onto $\mathbf{v}$ — $|M\mathbf{v}|^2$
- First singular vector — unit vector through origin that maximizes the sum of projections of all rows in M

$$\mathbf{v}_1 = \arg \max_{|\mathbf{v}|=1} |M\mathbf{v}|$$

- Second singular vector — unit vector through origin, perpendicular to $\mathbf{v}_1$, that maximizes the sum of projections of all rows in M

$$\mathbf{v}_2 = \arg \max_{\mathbf{v}\perp\mathbf{v}_1;\ |\mathbf{v}|=1} |M\mathbf{v}|$$

- Third singular vector — unit vector through origin, perpendicular to $\mathbf{v}_1$, $\mathbf{v}_2$, that maximizes the sum of projections of all rows in M

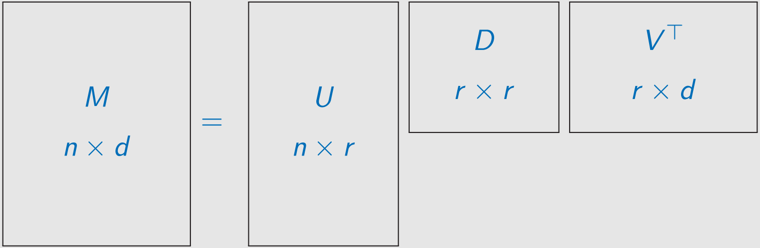
$$\mathbf{v}_3 = \arg \max_{\mathbf{v}\perp\mathbf{v}_1,\mathbf{v}_2;\ |\mathbf{v}|=1} |M\mathbf{v}|$$

Singular vectors ...

- With each singular vector $\mathbf{v}_j$, associated singular value is $\sigma_j = |M\mathbf{v}_j|$
- Repeat r times till $\max_{\mathbf{v} \perp \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r;\ |\mathbf{v}|=1} |M\mathbf{v}| = 0$
  - r turns out to be the rank of M
  - Vectors $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r\}$ are orthonormal right singular vectors
- Our greedy strategy provably produces "best-fit" dimension r subspace for M
  - Dimension r subspace that maximizes content of M projected onto it
- Corresponding left singular vectors are given by $\mathbf{u}_i = \dfrac{1}{\sigma_i} M\mathbf{v}_i$
- Can show that $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r\}$ are also orthonormal

Singular Value Decomposition

- M, dimension $n \times d$, of rank r uniquely decomposes as $M = UDV^\top$
  - $V = [\mathbf{v}_1\ \mathbf{v}_2\ \cdots\ \mathbf{v}_r]$ are the right singular vectors
  - D is a diagonal matrix with $D[i,i] = \sigma_i$, the singular values
  - $U = [\mathbf{u}_1\ \mathbf{u}_2\ \cdots\ \mathbf{u}_r]$ are the left singular vectors



Rank-k approximation

- M has rank r, SVD gives rank r decomposition
- Singular values are non-increasing — $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$
- Suppose we retain only k largest ones
- We have
  - Matrix of first k right singular vectors $V_k = [\mathbf{v}_1\ \mathbf{v}_2\ \cdots\ \mathbf{v}_k]$,
  - Corresponding singular values $\sigma_1, \sigma_2, \ldots, \sigma_k$
  - Matrix of k left singular vectors $U_k = [\mathbf{u}_1\ \mathbf{u}_2\ \cdots\ \mathbf{u}_k]$
- Let D_k be the $k \times k$ diagonal matrix with entries $\sigma_1, \sigma_2, \ldots, \sigma_k$
- Then $U_k D_k V_k^\top$ is the best fit rank-k approximation of M
- In other words, by truncating the SVD, we can focus on k most significant concepts implicit in M

Latent Semantic Indexing

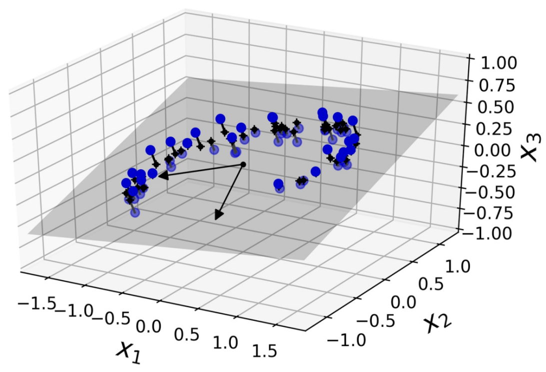
- Term-document matrix $M_{n \times d}$ with rank-k SVD $U_k D_k V_k^\top$
- $M_k = U_k D_k V_k^\top$ is the reduced term-document matrix
  - Column i of M_k is a document $\mathbf{d}_i$ over original terms
  - Column i of $V_k^\top$ is a transformed document $\widehat{\mathbf{d}}_i$
  - $\widehat{\mathbf{d}}_i$ is a representation of $\mathbf{d}_i$ in terms of k new abstract concepts
- $\mathbf{d}_i = U_k D_k \widehat{\mathbf{d}}_i$
- Computing backwards, $\widehat{\mathbf{d}}_i = D_k^{-1} U_k^{-1} \mathbf{d}_i$
- Columns of U are orthonormal $\Rightarrow$ $U^{-1} = U^\top$
- D_k is diagonal with entries $\sigma_i \Rightarrow D_k^{-1}$ is diagonal D_k' with entries $\dfrac{1}{\sigma_i}$
- Hence $\widehat{\mathbf{d}}_i = D_k' U_k^\top \mathbf{d}_i$

Query processing using LSI

- Given a query $\mathbf{q}$, represent in transformed space as $\widehat{\mathbf{q}}$
- Treating query as a document, apply the same transformation as for documents
  - $\widehat{\mathbf{d}}_i = D'_k U_k^\top \mathbf{d}_i$
  - $\widehat{\mathbf{q}} = D'_k U_k^\top \mathbf{q}$
- Now compare $\widehat{\mathbf{q}}$ with each $\widehat{\mathbf{d}}_i$ using cosine similarity
- Returned ranked list of documents

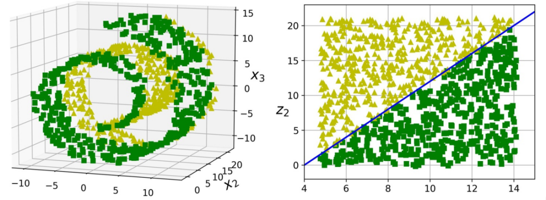
Dimensionality reduction

- In general, SVD allows us to work with a lower dimensional version of input
- Principal Component Anaylsis — transforms d-dimensional input to k-dimensional input by projecting on first k right singular vectors
- Example: PCA projection of blue points in 3D to black points in 2D

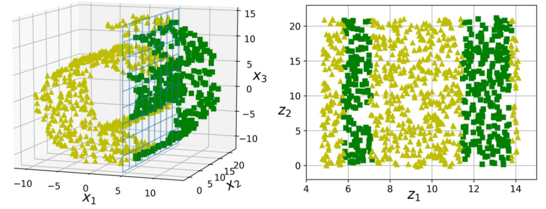


Dimensionality reduction . . .

- Unsupervised preprocessing technique — may make later steps easier, like simplifying classification boundaries
- Swiss roll dataset: dimensionality reduction helps



- Swiss roll dataset: dimensionality reduction does not help



Summary

- Singular Value Decomposition (SVD) finds best fit k-dimensional subspace for any matrix M
- In IR, it can help enhance the vector space model to handle problems like synonymy and polysemy — Latent Semantic Indexing
- Principal Component Analysis uses SVD for dimensionality reduction
- Unsupervised technique — often helps simplify the problem, but may not