Project 3: Information Retrieval

1. Suppose we have a dictionary that contains t terms and a database of d documents that are comprised of these terms. We seek to find the documents that are relevant to a given query. In the vector space model, we encode each document as a vector in $\mathbb{R}^t$, where its i-th component reflects how important the i-th term is in that document. The d vectors are stored as the columns $\vec{a}_j$ of $A \in \mathbb{R}^{t \times d}$, called the term-by-document matrix. The query, like the documents, is represented with a vector $\vec{x} \in \mathbb{R}^t$.

   Here, we set a_{ij} and x_i to 1 if the i-th term appears in the j-th document $\vec{a}_j$ and the query $\vec{x}$, and to 0 otherwise. In other words, all the terms that appear are considered equally important. For example, the first document "Nonlinear Optimization with Financial Applications" is represented by $\vec{a}_1$, whose 4th, 15th, 22nd, and 23rd entries equal to 1.

   Recall the formula for the angle θ_j between vectors $\vec{a}_j$ and $\vec{x}$:

   $$\text{(1-1)} \quad \cos\theta_j = \frac{\vec{a}_j^T \vec{x}}{||\vec{a}_j||_2 ||\vec{x}||_2}.$$

   The more similar the two vectors are, the closer $\cos\theta_j$ is to 1. In this exercise, we shall claim that the query matches the j-th document if $\cos\theta_j > 0.5$. (Note, multiplying $\vec{a}_j$ or $\vec{x}$ by a positive constant does not change the cosine value. Thus, we can normalize the document vectors or the query vector if we want to.)

   (a) Use the cosine value to decide if the document "Partial Differential Equations in Fluid Dynamics" matches the query "Linear Algebra and Differential Equations with Applications in Fluids."

   (b) Let $A = QR$ be the full QR factorization, with $Q \in \mathbb{R}^{t \times t}$ orthogonal and $R \in \mathbb{R}^{t \times d}$ upper-triangular. If the rank of A is r, we can partition

   $$A = \left[\begin{array}{c|c} Q_A & Q_A^\perp \end{array}\right] \left[\begin{array}{c} R_A \\ \hline 0 \end{array}\right] = Q_A R_A \ \textit{(reduced)},$$

   where $Q_A \in \mathbb{R}^{t \times r}$ and $R_A \in \mathbb{R}^{r \times d}$. ($Q_A^\perp$ indicates that its columns are orthogonal to those of Q_A.)

   From (1-1), show that we can instead compute the cosine value by

   $$\text{(1-2)} \quad \cos\theta_j = \frac{\vec{r}_{A,j}^T (Q_A^T \vec{x})}{||\vec{r}_{A,j}||_2 ||\vec{x}||_2},$$

   where $\vec{r}_{A,j} \in \mathbb{R}^r$ denotes the j-th column of R_A.

   (As an implementation detail, A is likely not full-rank, so we must do a QR factorization with column pivoting to get $AP = QR$, where P is a permutation matrix. The matrix R can be partitioned as

   $$R = \left[\begin{array}{c} R_A \\ \hline 0 \end{array}\right] \equiv \left[\begin{array}{c|c} R_{A,L} & R_{A,R} \\ \hline 0 & 0 \end{array}\right],$$

   where $R_{A,L} \in \mathbb{R}^{r \times r}$ is nonsingular and upper-triangular. We see that $\text{rank}(A) = \text{rank}(R_{A,L}) = r$. Furthermore, the diagonals of $R_{A,L}$ are decreasing in magnitude. In this exercise, the documents have been ordered so that permutation is not needed, i.e. $P = I$, and we just get $A = QR$.)

(c) Because $I = QQ^T = Q_A Q_A^T + Q_A^\perp (Q_A^\perp)^T$, we can decompose the query vector $\vec{x}$ into

$$\vec{x} = \vec{x}_A + \vec{x}_A^\perp \equiv Q_A Q_A^T \vec{x} + Q_A^\perp (Q_A^\perp)^T \vec{x},$$

where $\vec{x}_A = Q_A Q_A^T \vec{x}$ is the orthogonal projection of $\vec{x}$ onto the column space of Q_A.

Demonstrate that $\vec{x}_A$ is the best approximation of $\vec{x}$ in the column space of A, i.e.

$$||\vec{x} - \vec{x}_A||_2 = \min_{\vec{y} \in \text{col}(A)} ||\vec{x} - \vec{y}||_2.$$

Hint: With $\vec{y} \in \text{col}(A)$, consider $||\vec{x} - \vec{y}||_2^2 = ||\vec{x} - \vec{x}_A + \vec{x}_A - \vec{y}||_2^2$.

(d) We can use the decomposition $\vec{x} = \vec{x}_A + \vec{x}_A^\perp$ to also express the cosine value:

$$\text{(1-3)} \quad \cos\theta_j = \frac{\vec{a}_j^T \vec{x}_A}{||\vec{a}_j||_2 ||\vec{x}||_2} = \frac{\vec{r}_{A,j}^T (Q_A^T \vec{x}_A)}{||\vec{r}_{A,j}||_2 ||\vec{x}||_2}.$$

We see that the user's imperfect query $\vec{x}$ gets replaced in the inner product computation by its best approximation $\vec{x}_A$ from the database, col(A). The component $\vec{x}_A^\perp$, which does not share content with the database, is ignored. Based on this, we are motivated to consider a new measure of similiarity:

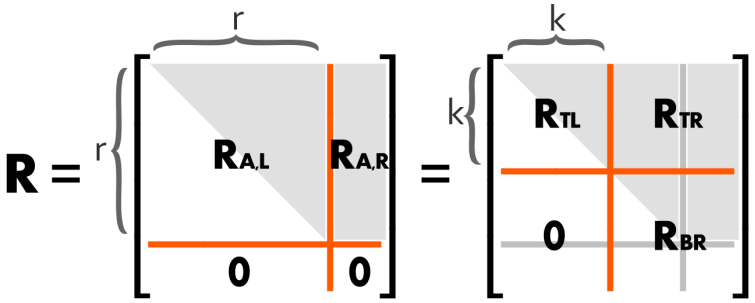
$$\text{(2)} \quad \cos\theta_j' = \frac{\vec{a}_j^T \vec{x}_A}{||\vec{a}_j||_2 ||\vec{x}_A||_2} = \frac{\vec{r}_{A,j}^T (Q_A^T \vec{x}_A)}{||\vec{r}_{A,j}||_2 ||\vec{x}_A||_2}.$$

Show that $\cos\theta_j \leq \cos\theta_j'$, i.e. while the new cosine value can identify more of the relevant documents, it may also return more irrelevant ones.

Hint: Show that $\cos\theta_j = \dfrac{||\vec{x}_A||_2}{\sqrt{||\vec{x}_A||_2^2 + ||\vec{x}_A^\perp||_2^2}} \cos\theta_j'$.

(e) The database may form over time, by many people with different experiences and different opinions on which terms are relevant in a given document and to what extent. Thus, the term-by-document matrix may be more accurate if we allow a perturbation $A + E$, where the entries of E reflect the uncertainties.

Now, if $A + E$ happens to have a lesser rank k, then we can argue that the database is really a rank-k matrix instead. To find E so that $rank(A + E) = k$, recall from part (b) that rank(A) = rank$(R_{A,L})$. We first repartition R so that



Note, $R_{TL} \in \mathbb{R}^{k \times k}$ is a submatrix of $R_{A,L}$ and R_{BR} contains some nonzero entries as a result.

Then, we let $A + E := Q\tilde{R}$, where $\tilde{R}$ is formed by zeroing out R_{BR}, i.e.

$$\tilde{\mathbf{R}} = \begin{bmatrix} \mathbf{R_{TL}} & \mathbf{R_{TR}} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

We see that $\text{rank}(A + E) = \text{rank}(R_{TL}) = k$.

Show that $||A||_F = ||R||_F$, and as a consequence,

$$\frac{||E||_F}{||A||_F} = \frac{||R_{BR}||_F}{||R||_F}.$$

Interpret in words what this means.

(f) Let us now compute the cosine values using the perturbed matrix $A + E$. Notice that we do not need to form E to work with the "$\vec{a}_j$" in the formula ($\vec{a}_j$ here stands for the j-th column of $A + E$). This is because we have the QR factorization of $A + E$, namely $A + E = Q\tilde{R}$.

If we partition

$$A + E = \left[\begin{array}{c|c} Q_{A+E} & Q_{A+E}^{\perp} \end{array}\right] \begin{bmatrix} R_{A+E} \\ \hline 0 \end{bmatrix} = Q_{A+E}R_{A+E} \text{ (reduced)},$$

so that $Q_{A+E} \in \mathbb{R}^{t\times k}$ and $R_{A+E} \equiv \left[\begin{array}{c|c} R_{TL} & R_{TR} \end{array}\right] \in \mathbb{R}^{k\times d}$, we can compute instead

$$\text{(3-2)} \quad \cos\theta_j = \frac{\vec{r}_{A+E,j}^T(Q_{A+E}^T\vec{x})}{||\vec{r}_{A+E,j}||_2||\vec{x}||_2},$$

$$\text{(4)} \quad \cos\theta_j' = \frac{\vec{r}_{A+E,j}^T(Q_{A+E}^T\vec{x}_{A+E})}{||\vec{r}_{A+E,j}||_2||\vec{x}_{A+E}||_2},$$

where $\vec{x}_{A+E} = Q_{A+E}Q_{A+E}^T\vec{x}$ is the orthogonal projection of $\vec{x}$ onto the column space of Q_{A+E}, and $\vec{r}_{A+E,j} \in \mathbb{R}^k$ is the j-th column of R_{A+E}.

Complete and run **project3.m** with the query "Linear Algebra and Differential Equations with Applications in Fluids." Verify that the same cosine value from part (a) is obtained, and remark on how good the matches are for each of the four options. Are there any misses or any false positives?

Note that $r = \text{rank}(A) = 21$, and we set $k = \text{rank}(A + E) = 16$. Using the formula in part (e), find how much relative change we have made from A to $A + E$.

Terms

1. Acoustics
2. Algebra, Algebraic
3. Analysis, Analytic
4. Application, Applications
5. Chaos
6. Complex
7. Control
8. Differential
9. Distribution
10. Dynamics
11. Economic, Economics
12. Element, Elements
13. Engineering
14. Equation, Equations
15. Finance, Financial
16. Finite
17. Fluid, Fluids
18. Geometry
19. Linear
20. Matrix, Matrices
21. Mechanics, Mechanical
22. Nonlinear
23. Optimization
24. Partial
25. Real
26. Statistics, Statistical
27. Stress
28. Structural
29. Theory, Theoretic, Theoretical
30. Transform, Transformations

Documents

1. **Nonlinear Optimization** with **Financial Applications**
2. **Finite Element** Modeling for **Stress Analysis**
3. An Introduction to **Mechanics**
4. **Matrices** and **Linear Transformations**
5. **Partial Differential Equations** in **Fluid Dynamics**
6. **Statistics** Fourth Edition
7. Mathematical **Control Theory**
8. Emerging **Applications** of **Algebraic Geometry**
9. **Real** and **Complex Analysis**
10. **Nonlinear Dynamics** and **Chaos**
11. **Theory** of **Nonlinear Acoustics** in **Fluids**
12. **Distribution Theory** and **Transform Analysis**
13. **Financial Engineering** with **Finite Elements**
14. **Linear Algebra** Done Right
15. **Differential Equations** Second Edition
16. **Real Analysis** with **Economic Applications**
17. **Linear** and **Nonlinear Structural Mechanics**
18. An Introduction to **Fluid Dynamics**
19. The **Finite Element** Methods
20. **Linear Algebra** and **Matrix Theory**
21. **Linear** and **Nonlinear Optimization**
22. **Partial Differential Equations** in **Mechanics**
23. The **Finite Element** Method for **Fluid Dynamics**
24. **Linear Algebra** Done Right Second Edition
25. **Statistical Mechanics**