Scientific Computation

Spring, 2019

Lecture 13

Dimension reduction

- **At the end of last lecture, briefly discussed *dimension reduction***
- **PCA transforms** A → G **using the eigenvectors of** AA^T**:** $U^T A = G$

where, $AA^T = USU^T$

The idea behind dimension reduction -- notice that:

$$A = UG = u_1\tilde{x}_1^T + u_2\tilde{x}_2^T + ...$$

and retain only the first p **components.** (u_i **corresponds to the** i^{th} **largest eigenvalue in** S**)**

The variance of these p **components corresponds to the sum of the first** p **eigenvalues in** S

Low-rank approximation

- **Dimension reduction is closely related to more general idea of a *low-rank* approximation a matrix**

- Rank(A) = **dimension of column space of** A **= dimension of row space of** A

- **You can find the rank of a matrix using the SVD:** $A = U\Sigma W^T$
  - **Here:** A **is** m x n **,** U **is** m x m**,** W **is** n x n
  - r **is the rank of** A **and is the number of non-zero eigenvalues of** AA^T **(and** A^TA**)**
  - Σ **is a** m x n **matrix with the singular values of** A **on its main diagonal**
    - r **of these are non-zero,**
- **And this motivates the idea of a low-rank approximation**

- **We (I) usually think of matrix multiplication as a sum of inner products**

$$A = UG = u_1\tilde{x}_1^T + u_2\tilde{x}_2^T + \ldots$$

- **But here, we think in terms of *outer* products, *e.g.***

Low-rank approximation

- Let's restate the SVD, $A = U\Sigma W^T$,

  as a sum of outer products: $A = \sigma_1 u_1 w_1^T + \sigma_2 u_2 w_2^T + ... + \sigma_r u_r w_r^T$

- The right-hand side is a sum of rank-1 matrices
  - Each column of $u_1 w_1^T$ is u_1 multiplied by a number

  - We want to truncate the RHS after p terms

    - This gives a rank-p approximation of A

    - How do we assess the quality of this approximation?

Low-rank approximation

- Let's restate the SVD, $A = U\Sigma W^T$,

  as a sum of outer products: $A = \sigma_1 u_1 w_1^T + \sigma_2 u_2 w_2^T + ... + \sigma_r u_r w_r^T$

- The right-hand side is a sum of rank-1 matrices
  - Each column of $u_1 w_1^T$ is u_1 multiplied by a number
  - We want to truncate the RHS after p terms
    - This gives a rank-p approximation of A
    - How do we assess the quality of this approximation, A_p?
  - We'll work with the *Frobenius norm* of a matrix: $|A|_F = \sum_{i=1}^{M}\sum_{j=1}^{N}(A_{ij})^2$
  - And for a 1-term approximation, we have:

$$|A_1|_F = trace(w_1 w_1^T) = \sigma_1^2$$

Low-rank approximation

- **So, the singular values indicate the approximation quality (and error) of a SVD-based low-rank approximation:** $|A_1|_F = \sigma_1^2 \sum_{i=1}^{N} \sum_{j=1}^{N} (w_i w_j) = \sigma_1^2$

- **Here, I've used the orthonormality of** w **and:** $|A|_F = \sum_{i=1}^{M} \sum_{j=1}^{N} (A_{ij})^2 = trace(A^T A) = \sum_{i=1}^{r} \sigma_i^2$

It's a little tedious to show, but we can generalize the above result as you would guess: $|A_p|_F = \sum_{i=1}^{p} \sigma_i^2$

And the 1st-p singular values indicate quality of a rank-p approximation

Low-rank approximation

- **In fact, it can be shown that the SVD-based rank-p approximation is the *best* low-rank approximation**

- **Problem definition: Find** B **such that** $|A - B|_F$ **is minimized and rank(**B**)=p**

- **Solution (Eckart-Young theorem):** $B = \sigma_1 u_1 w_1^T + \sigma_2 u_2 w_2^T + \ldots + \sigma_p u_p w_p^T$

  with: $A = U\Sigma W^T$

  and the approximation error is: $|A - B|_F = \sqrt{\sum_{i=p+1}^{rank(A)} \sigma_i^2}$

- **This result provides foundation for both dimensionality reduction from PCA and our "image compression" example**

- **And is important for our next example…**

Data analysis

- With PCA, the goal was to extract information from a dataset
- Now, we want to think about filling in information *missing* from a dataset
- There are many different approaches, we will look at just one based on low-rank matrix factorization
- Is this actually useful? Very important in computer graphics, vision, machine learning, and recommender systems
- We will focus on the latter example

Recommender systems

- How do Netflix, Amazon, Facebook, etc… decide what to recommend to you?
- They collect:
  - information about what you like (and dislike)
  - information about what everyone else likes
- And then attempt to predict what you will spend time and/or money on
- How is this data organized? – one example is a ratings matrix

Recommender systems

- **How is this data organized? – one example is a ratings matrix**

	Suits	Sex Education	Friends	Stranger Things	Killing Eve
Don	5	?	1	?	?
Liz	0	4	5	?	?
Kamala	2	?	3	?	5
Beto	1	5	4	5	?

- **How do we fill in the missing entries?**

- **Organizing idea: Need ratings for high-level concepts (e.g. genre) rather than individual movies**
  - **Note: in practice, ratings matrices are much, much larger**