SI231b: Matrix Computations

Lecture 21: Low-rank Approximation and Regularized Least Square

Yue Qiu

qiuyue@shanghaitech.edu.cn

School of Information Science and Technology

ShanghaiTech University

Nov. 25, 2021

Motivation Example: Image Compression

Original Image

- Let $\mathrm{A} \in \mathbb{R}^{m\times n}$ be a matrix whose $(i,\ j)$th entry a_{ij} represents the (i, j)th pixel of an image.
- memory consumption for storing A: $m * n$

Compressed Image

- using truncated SVD of A: store $\{\mathrm{u}_i, \sigma_i \mathrm{v}_i\}_{i=1}^{k}$ instead of the full A.
- the compressed image is represented by $\mathrm{B} = \sum_i^k \sigma_i \mathrm{u}_i \mathrm{v}_i^T$
- memory consumption for truncated SVD: $(m+n) * k$
  - much less than $m * n$ if $k \ll \min\{m, n\}$

original image, sizes 470×641

Figure 1: original image

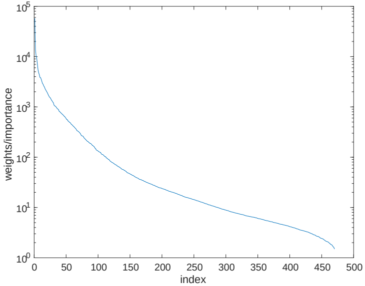


Figure 2: singular values

Image Compression Illustration

compressed image with $r = 10$

compressed image with $r = 20$

compressed image with $r = 30$

compressed image with $r = 40$

Aim: given a matrix $A \in \mathbb{R}^{m \times n}$ and an integer k with $0 \leq k \leq \text{rank}(A)$, find a matrix $B \in \mathbb{R}^{m \times n}$ such that $\text{rank}(B) \leq k$ and B best approximates A

- it is somehow unclear about what a "best approximation" means, and we will specify one later
- applications: PCA, dimensionality reduction, $\cdots\cdots$ the same kind of applications in matrix factorization
- truncated SVD: denote

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$$

  where the kth "partial sum" captures as much of the energy of A as possible, and the meaning of "energy" will be specified later
- then perform the aforementioned approximation by choosing $B = A_k$

Low-rank Approximation

Truncated SVD provides the best approximation in the LS sense:

Theorem[Eckart-Young-Mirsky]. Consider the following problem

$$\min_{\mathrm{B}\in\mathbb{R}^{m\times n},\ \ \mathrm{rank}(\mathrm{B})\leq k} \|\mathrm{A}-\mathrm{B}\|_F^2$$

where $\mathrm{A}\in\mathbb{R}^{m\times n}$ and $k\in\{1,\ldots,p\}$ with $p=\min\{m,\ n\}$ are given. The truncated SVD A_k is an optimal solution to the above problem and the minimum is $\sum_{i=k+1}^{p}\sigma_i^2$

- also note the matrix 2-norm version of the Eckart-Young-Mirsky theorem:

  $$\min_{\mathrm{B}\in\mathbb{R}^{m\times n},\ \ \mathrm{rank}(\mathrm{B})\leq k} \|\mathrm{A}-\mathrm{B}\|_2^2$$

  The truncated SVD A_k is an optimal solution to the above problem and the minimum is σ_{k+1}^2

  (cf. Theorem 2.4.8 in [Golub & van Loan 13'])

- the energy mentioned before is defined by either the Frobenius norm or the 2-norm

In practice, we are more interested in the factorized form of low-rank approximation,

$$\min_{\mathrm{A}\in\mathbb{R}^{m\times k},\mathrm{B}\in\mathbb{R}^{k\times n}} \|\mathrm{Y}-\mathrm{AB}\|_F^2$$

where $k \leq \min\{m,n\}$; A denotes a basis matrix; B is the coefficient matrix.

- the matrix factorization problem may be reformulated as (verify)

  $$\min_{\mathrm{Z}\in\mathbb{R}^{m\times n},\mathrm{rank}(\mathrm{Z})\leq k} \|\mathrm{Y}-\mathrm{Z}\|_F^2,$$

  and the truncated SVD $\mathrm{Y}_k = \sum_{i=1}^{k} \sigma_i \mathrm{u}_i \mathrm{v}_i^T$, where $\mathrm{Y} = \mathrm{U}\boldsymbol{\Sigma}\mathrm{V}^T$ denotes the SVD of Y, is an optimal solution by the Eckart-Young-Mirsky theorem.
- thus, an optimal solution to the matrix factorization problem is given by

  $$\mathrm{A} = [\ \mathrm{u}_1, \ldots, \mathrm{u}_k\ ], \qquad \mathrm{B} = [\ \sigma_1\mathrm{v}_1, \ldots, \sigma_k\mathrm{v}_k\ ]^T$$

Singular Value Inequalities

Similar to variational characterization of eigenvalues of real symmetric matrices, we can derive various variational characterization results for singular values, e.g.,

- Courant-Fischer characterization:

$$\sigma_k(\mathrm{A}) = \min_{\dim \mathcal{S}_{n-k+1} \subseteq \mathbb{R}^n} \; \max_{\mathrm{x} \in \mathcal{S}_{n-k+1}, \; \|\mathrm{x}\|_2 = 1} \|\mathrm{A}\mathrm{x}\|_2$$

- Weyl's inequality: for any $\mathrm{A}, \mathrm{B} \in \mathbb{R}^{m \times n}$,

$$\sigma_{k+l-1}(\mathrm{A}+\mathrm{B}) \leq \sigma_k(\mathrm{A}) + \sigma_l(\mathrm{B}), \qquad k, l \in \{1, \dots, p\}, \; k+l-1 \leq p.$$

  Also, note the corollaries

  - $\sigma_k(\mathrm{A}+\mathrm{B}) \leq \sigma_k(\mathrm{A}) + \sigma_1(\mathrm{B})$, $k = 1, \dots, p$
  - $|\sigma_k(\mathrm{A}+\mathrm{B}) - \sigma_k(\mathrm{A})| \leq \sigma_1(\mathrm{B})$, $k = 1, \dots, p$ (important results of perturbation theory)

- and many more...

Applying Weyl's inequality

- for any B with $\text{rank}(\mathrm{B}) \le k$, we have
  - $\sigma_l(\mathrm{B}) = 0$ for $l > k$
  - (Weyl) $\sigma_{i+k}(\mathrm{A}) \le \sigma_i(\mathrm{A}-\mathrm{B}) + \sigma_{k+1}(\mathrm{B}) = \sigma_i(\mathrm{A}-\mathrm{B})$ for $i = 1, \ldots, p-k$
  - and consequently

$$\|\mathrm{A}-\mathrm{B}\|_F^2 = \sum_{i=1}^{p} \sigma_i(\mathrm{A}-\mathrm{B})^2 \ge \sum_{i=1}^{p-k} \sigma_i(\mathrm{A}-\mathrm{B})^2 \ge \sum_{i=k+1}^{p} \sigma_i(\mathrm{A})^2$$

- the equality above is attained if we choose $\mathrm{B} = \mathrm{A}_k$

Advantages of Using Low-rank Factorized Form

Let $\mathrm{A} \in \mathbb{R}^{m\times n}$ being approximated by $\mathrm{B} = \mathrm{U}\mathrm{V}^T$ with $\mathrm{U} \in \mathbb{R}^{m\times r_k}$, $\mathrm{V} \in \mathbb{R}^{n\times r_k}$ and $r_k \ll \{m,\ n\}$, i.e., $\mathrm{B} \approx \mathrm{A}$.

Computational Complexity Reduction

- matrix-vector product with $\mathrm{z} \in \mathbb{R}^n$
  - $\mathcal{O}(mn)$ for Az
  - $\mathcal{O}\left(r_k(m+n)\right)$ for Bz
- matrix-matrix product with $\mathrm{Z} \in \mathbb{R}^{n\times n}$
  - $\mathcal{O}(mn^2)$ for AZ
  - $\mathcal{O}\left(r_k(m+n)n\right)$ for BZ

Memory Consumption Reduction

- $\mathcal{O}(mn)$ for A
- $\mathcal{O}\left(r_k(m+n)\right)$ for B

The key of low-rank approximation lies in the fact that

- all computations should be performed using low-rank factors U and V rather than the excplicit $\mathrm{B} = \mathrm{U}\mathrm{V}^T$
- the rank $r_k \ll \{m,\ n\}$

Rank Growth

In computations, to keep the results in factorized form, the rank will increase. For example, for $m \times n$ matrices $\mathrm{B} = \mathrm{U}_1\mathrm{V}_1^T$, $\mathrm{C} = \mathrm{U}_2\mathrm{V}_2^T$ and to compute $\mathrm{B} + \mathrm{C}$, we have

$$\mathrm{D} = \mathrm{B} + \mathrm{C} = \mathrm{U}_b\mathrm{V}_b^T + \mathrm{U}_c\mathrm{V}_c^T = \underbrace{\begin{bmatrix} \mathrm{U}_b & \mathrm{U}_c \end{bmatrix}}_{\mathrm{U}_d} \underbrace{\begin{bmatrix} \mathrm{V}_b^T \\ \mathrm{V}_c^T \end{bmatrix}}_{\mathrm{V}_d^T}.$$

The rank of D turns to be $r_b + r_c$ in the general case and continues growing when more computations are performed.

We need to reduce the rank for less computational complexity.

Rank Reduction

Keeping the rank bounded is the key in applying low-rank approximation for computations.

For an $m \times n$ matrix $\mathrm{A} = \mathrm{U}\mathrm{V}^T$ with low-rank factors $\mathrm{U} \in \mathbb{R}^{m\times r}$ and $\mathrm{V} \in \mathbb{R}^{n\times r}$, the following procedure returns a best rank r' of A with $r' < r$

1. compute a reduced QR factorization of U, i.e., $\mathrm{U} = \mathrm{Q}\mathrm{R}$ with $\mathrm{Q} \in \mathbb{R}^{m\times r}$ and $\mathrm{R} \in \mathbb{R}^{r\times r}$ $\quad (\mathcal{O}(r^2 m)\ \texttt{cost})$
2. form $\mathrm{C} = \mathrm{R}\mathrm{V}^T$ with $\mathrm{C} \in \mathbb{R}^{r\times n}$ $\quad (\mathcal{O}(r^2 n)\ \texttt{cost})$
3. compute the SVD of C, i.e., $\mathrm{C} = \begin{bmatrix} \mathrm{U}_c^{(1)} & \mathrm{U}_c^{(2)} \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma}_c^{(1)} & \\ & \mathbf{\Sigma}_c^{(2)} \end{bmatrix} \begin{bmatrix} (\mathrm{V}_c^{(1)})^T \\ (\mathrm{V}_c^{(2)})^T \end{bmatrix}$ with $\mathrm{U}_c^{(1)}$ having r' columns $\quad (\mathcal{O}(r^2 n)\ \texttt{cost})$
4. $\tilde{\mathrm{A}} = \mathrm{Q}\mathrm{U}_c^{(1)}\mathbf{\Sigma}_c^{(1)}(\mathrm{V}_c^{(1)})^T$ returns the best rank r' approximation of A

Can you prove the optimality?

We have seen from the previous analysis that the key to keep the computational complexity low using low-rank approximation is

- using low-rank factorized form
- reducing the increased rank while performing computations

To perform computations using low-rank approximations, we need to start with low-rank factorized form,

- may be already given
- using SVD to compute (one time cost)
- using randomized algorithm to find one if SVD is too expensive, cf. the following reference by Caltech
  - N. Halko, P. G. Martinsson, and J. A. Tropp. Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions. *SIAM Review*, vol. 53, pp. 217–288, 2011.

Other Related Low-rank Approximation

We have introduced the low-rank approximation using SVD in this lecture, which in turn gives optimal results. Other related low-rank approximation methods which are less accurate but computationally cheaper include

- CUR factorization $A \approx CUR$ where C is from columns of A, R contains rows of A;
- skelton/cross approximation;
- nonnegative matrix factorization (NMF) (widely used in NLP)

For high dimensional data, tensor computations are used.

Sensitivity to Noise

Question: how sensitive is the LS solution when there is noise?

$$y = A\bar{x} + \boldsymbol{\nu},$$

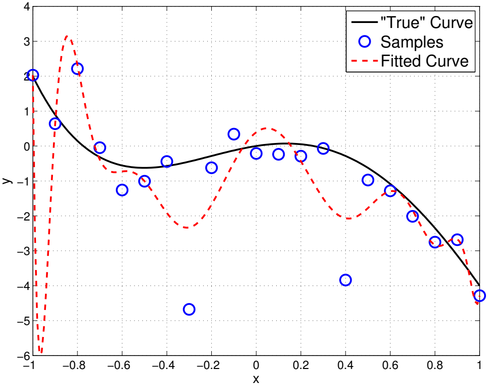
where $\bar{x}$ is the true result; $A \in \mathbb{R}^{m \times n}$ has full column rank; $\boldsymbol{\nu}$ is noise, modeled as a random vector, for example with mean zero and covariance $\gamma^2 I$ (white noise).

Mean square error (MSE) analysis: from $x_{LS} = A^{\dagger} y = \bar{x} + A^{\dagger}\boldsymbol{\nu}$ we get

$$\begin{aligned} \mathrm{E}[\|x_{LS} - \bar{x}\|_2^2] &= \mathrm{E}[\|A^{\dagger}\boldsymbol{\nu}\|_2^2] = \mathrm{E}[\mathrm{tr}(A^{\dagger}\boldsymbol{\nu}\boldsymbol{\nu}^T(A^{\dagger})^T)] = \mathrm{tr}(A^{\dagger}\mathrm{E}[\boldsymbol{\nu}\boldsymbol{\nu}^T](A^{\dagger})^T] \\ &= \gamma^2 \mathrm{tr}(A^{\dagger}(A^{\dagger})^T) \\ &= \gamma^2 \sum_{i=1}^{n} \frac{1}{\sigma_i^2(A)} \end{aligned}$$

Observation: the MSE becomes very large if some $\sigma_i(A)$'s are close to zero.

Example: Curve Fitting



The same curve fitting example in Lecture 7. The "true" curve is the true $f(x)$ with polynomial order $n = 4$. In practice, the model order may not be known and we may have to guess. The fitted curve above is done by LS with a guessed model order $n = 16$.

Intuition: replace $x_{LS} = (A^TA)^{-1}A^Ty$ by

$$x_{RLS} = (A^TA + \lambda I)^{-1}A^Ty,$$

for some $\lambda > 0$, where the term λI is added to improve the conditioning of the system , i.e., move the singular values of A^TA away from zero, thereby attempting to reduce noise sensitivity.

How may we make sense out of such a modification?

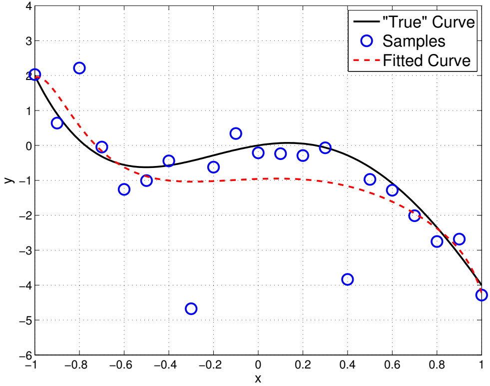
ℓ_2-regularized LS: find an x that solves

$$\min_{x \in \mathbb{R}^n} \|Ax - y\|_2^2 + \lambda \|x\|_2^2$$

for some predetermined $\lambda > 0$.

- the solution is uniquely given by $x_{RLS} = (A^TA + \lambda I)^{-1}A^Ty$
- the formulation says that we try to minimize both $\|y - Ax\|_2^2$ and $\|x\|_2^2$, and λ controls which one should be more emphasized in the minimization

Example: Curve Fitting Using ℓ_2-Regularization



The fitted curve is done by ℓ_2-regularized LS with a guessed model order $n = 18$ and with $\lambda = 0.1$.

Readings

If you are interested in the modified least squares problems and the their solution via SVD, you are suggested to read

- Gene H. Golub and Charles F. Van Loan. Matrix Computations, *Johns Hopkins University Press*, 2013.

  Chapter 6.1 – 6.4.