Singular Value Decomposition and Linear Least Squares
CSE 6367: Computer Vision

Instructor: William J. Beksi

Introduction

- The **singular value decomposition** (SVD) efficiently organizes any given matrix

Introduction

- The **singular value decomposition** (SVD) efficiently organizes any given matrix

- It is useful for several purposes:

Introduction

- The **singular value decomposition** (SVD) efficiently organizes any given matrix

- It is useful for several purposes:

  - It explains clearly the action of the matrix (as a linear operator) and it can be used to reestablish many elementary properties of matrices

Introduction

- The **singular value decomposition** (SVD) efficiently organizes any given matrix

- It is useful for several purposes:

  - It explains clearly the action of the matrix (as a linear operator) and it can be used to reestablish many elementary properties of matrices

  - It can efficiently compress a matrix with the least loss of information (in a **least squares** sense) by summarizing only the most relevant features and suppressing unnecessary information (e.g. SVD is used for dimensionality reduction by projecting a dataset onto a lower dimensional subspace)

Eigenvalue Decomposition of Symmetric Matrices

- The **eigenvalue decomposition** (EVD) of a symmetric matrix $A \in \mathbb{R}^{n \times n}$ is

$$A = U \Lambda U^T \tag{1}$$

where $\Lambda \in \mathbb{R}^{n \times n}$ is diagonal, $U \in \mathbb{R}^{n \times n}$, and $U U^T = I$ (i.e. U is orthogonal)

Eigenvalue Decomposition of Symmetric Matrices

- The **eigenvalue decomposition** (EVD) of a symmetric matrix $A \in \mathbb{R}^{n \times n}$ is

$$A = U \Lambda U^T \tag{1}$$

  where $\Lambda \in \mathbb{R}^{n \times n}$ is diagonal, $U \in \mathbb{R}^{n \times n}$, and $UU^T = I$ (i.e. U is orthogonal)

- We denote the eigenvalues of $\Lambda_{n \times n}$ by $\lambda_1, \ldots, \lambda_n$ and the columns of U by $\mathbf{u}_1, \ldots, \mathbf{u}_n$

EVD Formulation

- Let $\{\mathbf{u}_i\}_{i=1}^n, \forall\, 1 \leq i \leq n$ be an orthonormal basis for $\mathbb{R}^n$, then the EVD can be formulated as

$$A\mathbf{u}_i = \lambda_i \mathbf{u}_i \qquad (2)$$

where $\lambda_i \in \mathbb{R}$

EVD Formulation

- Thus, $A = U\Lambda U^T$ iff $AU = U\Lambda$, that is

$$A \begin{bmatrix} \mathbf{u}_1 & \ldots & \mathbf{u}_n \end{bmatrix} = \begin{bmatrix} \mathbf{u}_1 & \ldots & \mathbf{u}_n \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}$$

or equivalently

$$\begin{bmatrix} A\mathbf{u}_1 & \ldots & A\mathbf{u}_n \end{bmatrix} = \begin{bmatrix} \lambda_1\mathbf{u}_1 & \ldots & \lambda_n\mathbf{u}_n \end{bmatrix}$$

EVD Formulation

- Thus, $A = U \Lambda U^T$ iff $AU = U\Lambda$, that is

$$A \begin{bmatrix} \mathbf{u}_1 & \ldots & \mathbf{u}_n \end{bmatrix} = \begin{bmatrix} \mathbf{u}_1 & \ldots & \mathbf{u}_n \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}$$

or equivalently

$$\begin{bmatrix} A\mathbf{u}_1 & \ldots & A\mathbf{u}_n \end{bmatrix} = \begin{bmatrix} \lambda_1 \mathbf{u}_1 & \ldots & \lambda_n \mathbf{u}_n \end{bmatrix}$$

- $\{\mathbf{u}_i\}_{i=1}^{n}$ is an orthonormal basis for $\mathbb{R}^n$, and is equivalent with the fact that U is an orthogonal matrix

EVD Formulation

- Another equivalent formulation of the EVD is

$$A = \sum_{i=1}^{n} \lambda_i \mathbf{u}_i \mathbf{u}_i^T \tag{3}$$

where $\{\mathbf{u}_i\}_{i=1}^n$ is as in (2) and $\{\lambda_i\}_{i=1}^n \subset \mathbb{R}$

Proof of the EVD Formulation Equivalence

- To prove the equivalence of (1) and (3) we use the following basic identity for multiplying a row of column vectors with a column of row vectors

$$\begin{bmatrix} \mathbf{v}_1 & \dots & \mathbf{v}_n \end{bmatrix} \begin{bmatrix} \mathbf{u}_1^T \\ \vdots \\ \mathbf{u}_n^T \end{bmatrix} = \sum_{i=1}^{n} \mathbf{v}_i \mathbf{u}_i^T \qquad (4)$$

Proof of the EVD Formulation Equivalence

- To prove the equivalence of (1) and (3) we use the following basic identity for multiplying a row of column vectors with a column of row vectors

$$
\begin{bmatrix} \mathbf{v}_1 & \dots & \mathbf{v}_n \end{bmatrix} \begin{bmatrix} \mathbf{u}_1^T \\ \vdots \\ \mathbf{u}_n^T \end{bmatrix} = \sum_{i=1}^{n} \mathbf{v}_i \mathbf{u}_i^T \tag{4}
$$

- We express the equality in (1) as follows

$$
A = \begin{bmatrix} \mathbf{u}_1 & \dots & \mathbf{u}_n \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \begin{bmatrix} \mathbf{u}_1^T \\ \vdots \\ \mathbf{u}_n^T \end{bmatrix} \tag{5}
$$

Proof of the EVD Formulation Equivalence

- By applying direct multiplication with a diagonal matrix and then the identity (4), we expand the expression in (5) as follows

$$A = \begin{bmatrix} \lambda_1 \mathbf{u}_1 & \ldots & \lambda_n \mathbf{u}_n \end{bmatrix} \begin{bmatrix} \mathbf{u}_1^T \\ \vdots \\ \mathbf{u}_n^T \end{bmatrix} = \sum_{i=1}^{n} \lambda_i \mathbf{u}_i \mathbf{u}_i^T$$

and this establishes the equivalence of (1) and (3)

Eigenvalues of Symmetric Matrices

- The eigenvalues of symmetric matrices are real, however they can be negative

Eigenvalues of Symmetric Matrices

- The eigenvalues of symmetric matrices are real, however they can be negative

- In the special case where a symmetric matrix has the form $A^T A$ (A is a rectangular matrix) the eigenvalues are nonnegative

Eigenvalues of Symmetric Matrices

- The eigenvalues of symmetric matrices are real, however they can be negative

- In the special case where a symmetric matrix has the form $A^T A$ (A is a rectangular matrix) the eigenvalues are nonnegative

- Thus if $A^T A \mathbf{v} = \lambda \mathbf{v}$, then $\mathbf{v}^T A^T A \mathbf{v} = \lambda \mathbf{v}^T \mathbf{v}$ and consequently

$$\lambda = \frac{\mathbf{v}^T A^T A \mathbf{v}}{\mathbf{v}^T \mathbf{v}} = \frac{||A\mathbf{v}||_2^2}{||\mathbf{v}||_2^2} = ||A\mathbf{v}||_2 \geq 0$$

Eigenvalues of Symmetric Matrices

- The eigenvalues of symmetric matrices are real, however they can be negative

- In the special case where a symmetric matrix has the form $A^T A$ (A is a rectangular matrix) the eigenvalues are nonnegative

- Thus if $A^T A \mathbf{v} = \lambda \mathbf{v}$, then $\mathbf{v}^T A^T A \mathbf{v} = \lambda \mathbf{v}^T \mathbf{v}$ and consequently

$$\lambda = \frac{\mathbf{v}^T A^T A \mathbf{v}}{\mathbf{v}^T \mathbf{v}} = \frac{||A\mathbf{v}||_2^2}{||\mathbf{v}||_2^2} = ||A\mathbf{v}||_2 \geq 0$$

- Recall that the Euclidean norm of a vector $\mathbf{y} \in \mathbb{R}^k$, $k \in \mathbb{N}$ is $||\mathbf{y}||_2 = \sqrt{\mathbf{y}^T \mathbf{y}}$

The Thin SVD

- Let $A \in \mathbb{R}^{m \times n}$ and assume that $m \geq n$ (otherwise we consider A^T), then the **thin** SVD of A is

$$A_{m \times n} = U_{m \times n} \Sigma_{n \times n} V_{n \times n}^T$$

where Σ is a diagonal matrix with nonnegative elements, V is orthogonal (i.e. $VV^T = V^TV = I$) and U satisfies $U_{m \times n}^T U_{m \times n} = I_{n \times n}$ iff $m = n$

The Thin SVD

- We refer to the diagonal elements of Σ, denoted by $\{\sigma_i\}_{i=1}^n$, as the singular values of A and order them so that
  $$\sigma_1 \geq \cdots \geq \sigma_n \geq 0$$

The Thin SVD

- We refer to the diagonal elements of Σ, denoted by $\{\sigma_i\}_{i=1}^n$, as the singular values of A and order them so that $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$

- We denote the columns of V by $\mathbf{v}_1, \ldots, \mathbf{v}_n$ and refer to them as right singular vectors

The Thin SVD

- We refer to the diagonal elements of Σ, denoted by $\{\sigma_i\}_{i=1}^n$, as the singular values of A and order them so that $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$

- We denote the columns of V by $\mathbf{v}_1, \ldots, \mathbf{v}_n$ and refer to them as right singular vectors

- Similarly, the columns of U are denoted by $\mathbf{u}_1, \ldots, \mathbf{u}_n$ and are called left singular vectors

The Full SVD

- The **full** SVD of A takes the form

$$A_{m \times n} = U_{m \times m} \Sigma_{m \times n} V_{n \times n}^T$$

where U and V are orthogonal matrices and $\Sigma_{m \times n}$ is obtained from $\Sigma_{n \times n}$ of the thin SVD by padding it with an $(m - n) \times n$ bock of zeros, i.e.

$$\Sigma_{m \times n} = \begin{bmatrix} \Sigma_{n \times n} \\ \mathbf{0}_{(m-n) \times n} \end{bmatrix}$$

The Compressed SVD

- The **compressed** SVD is even thinner than the thin SVD

The Compressed SVD

- The **compressed** SVD is even thinner than the thin SVD

- Let $r = r(A)$ be the number of nonzero singular values of A (note that $\sigma_1 \geq \ldots \geq \sigma_r > \sigma_{r+1} = \ldots = \sigma_n = 0$), the compressed SVD of A has the form

$$A_{m \times n} = U_{m \times r} \Sigma_{r \times r} V_{n \times r}^T$$

where $\Sigma_{r \times r}$ is diagonal (with positive elements), $U_{m \times r}^T U_{m \times r} = I_{r \times r}$ and $V_{n \times r}^T V_{n \times r} = I_{r \times r}$

The Full, Thin, and Compressed SVDs

- The full SVD is common in pure mathematics (due to its use of two orthogonal transformations) and is the default for the MATLAB `svd` command

The Full, Thin, and Compressed SVDs

- The full SVD is common in pure mathematics (due to its use of two orthogonal transformations) and is the default for the MATLAB `svd` command

- In practice the thin SVD is more economical and avoids redundant space, it can be obtained from the MATLAB `svd` command by choosing the option 0

The Full, Thin, and Compressed SVDs

- The full SVD is common in pure mathematics (due to its use of two orthogonal transformations) and is the default for the MATLAB `svd` command

- In practice the thin SVD is more economical and avoids redundant space, it can be obtained from the MATLAB `svd` command by choosing the option 0

- The compressed SVD is even more economical, however the problem is that it is difficult to precisely determine r (a smaller estimate for r may be used)

SVD Formulation (Thin)

- Similar to the three equivalent formulations of the EVD, the SVD has three equivalent formulations

SVD Formulation (Thin)

- Similar to the three equivalent formulations of the EVD, the SVD has three equivalent formulations

- For simplicity, we formulate them for the thin SVD

$$A_{m \times n} = U_{m \times n} \Sigma_{n \times n} V_{n \times n}^T \tag{6}$$

where Σ is diagonal with nonnegative elements, $U^T U = I_{n \times n}$, and $V^T V = I_{n \times n}$

SVD Formulation (Thin)

- Let $\{\mathbf{v}_i\}_{i=1}^n$, $i = 1, \ldots, n$ be an orthonormal basis for $\mathbb{R}^n$ and $\{\mathbf{u}\}_{i=1}^n$ be an orthonormal system for $\mathbb{R}^m$, $\sigma_1 \geq \ldots \geq \sigma_n \geq 0$, then

$$A\mathbf{v}_i = \sigma_i \mathbf{u}_i \tag{7}$$

and

$$A = \sum_{i=1}^n \sigma_i \mathbf{u}_i \mathbf{v_i}^T \tag{8}$$

where $\{\mathbf{u}_i\}_{i=1}^n$, $\{\mathbf{v}_i\}_{i=1}^n$, and $\{\sigma_i\}_{i=1}^n$ are as in (2)

SVD Formulation (Thin)

- Let $\{\mathbf{v}_i\}_{i=1}^{n}, i = 1, \ldots, n$ be an orthonormal basis for $\mathbb{R}^n$ and $\{\mathbf{u}\}_{i=1}^{n}$ be an orthonormal system for $\mathbb{R}^m$, $\sigma_1 \geq \ldots \geq \sigma_n \geq 0$, then

$$A\mathbf{v}_i = \sigma_i \mathbf{u}_i \tag{7}$$

and

$$A = \sum_{i=1}^{n} \sigma_i \mathbf{u_i} \mathbf{v_i}^T \tag{8}$$

where $\{\mathbf{u}_i\}_{i=1}^{n}$, $\{\mathbf{v}_i\}_{i=1}^{n}$, and $\{\sigma_i\}_{i=1}^{n}$ are as in (2)

- The equivalence of (6)-(8) is proved similarly to the equivalence of (1)-(3)

Example: Computing the SVD

- Let

$$A_{3\times 2} = \begin{bmatrix} 2 & -1 \\ 1 & 2 \\ 1 & 1 \end{bmatrix}$$

and form $A^T A = \begin{bmatrix} 6 & 1 \\ 1 & 6 \end{bmatrix}$

Example: Computing the SVD

- Let

$$A_{3\times 2} = \begin{bmatrix} 2 & -1 \\ 1 & 2 \\ 1 & 1 \end{bmatrix}$$

and form $A^T A = \begin{bmatrix} 6 & 1 \\ 1 & 6 \end{bmatrix}$

- Since the trace of this matrix is 12 and the determinant is 35, its eigenvalues are $\lambda_1 = 7$ and $\lambda_2 = 5$ and therefore $\sigma_1 = \sqrt{7}$ and $\sigma_2 = \sqrt{5}$

Example: Computing the SVD

- Next, we need to compute the eigenvectors of $A^T A$

Example: Computing the SVD

- Next, we need to compute the eigenvectors of $A^T A$

- One may quickly notice that $[1, 1]^T$ is an eigenvector with eigenvalue 7, the other eigenvector needs to be orthogonal to it and thus $[1, -1]^T$ is an eigenvector with eigenvalue 5

Example: Computing the SVD

- Next, we need to compute the eigenvectors of $A^T A$

- One may quickly notice that $[1, 1]^T$ is an eigenvector with eigenvalue 7, the other eigenvector needs to be orthogonal to it and thus $[1, -1]^T$ is an eigenvector with eigenvalue 5

- To form an orthonormal basis we normalize these vectors and obtain $\mathbf{v}_1 = [1/\sqrt{2}, 1/\sqrt{2}]^T$ and $\mathbf{v}_2 = [1/\sqrt{2}, -1/\sqrt{2}]^T$

Example: Computing the SVD

- Next, we need to compute the eigenvectors of $A^T A$

- One may quickly notice that $[1, 1]^T$ is an eigenvector with eigenvalue 7, the other eigenvector needs to be orthogonal to it and thus $[1, -1]^T$ is an eigenvector with eigenvalue 5

- To form an orthonormal basis we normalize these vectors and obtain $\mathbf{v}_1 = [1/\sqrt{2}, 1/\sqrt{2}]^T$ and $\mathbf{v}_2 = [1/\sqrt{2}, -1/\sqrt{2}]^T$

- Finally, $\mathbf{u}_1 = \frac{A\mathbf{v}_1}{\sigma_1} = [1/\sqrt{14}, 3/\sqrt{14}, 2/\sqrt{14}]^T$ and $\mathbf{u}_2 = \frac{A\mathbf{v}_2}{\sigma_2} = [3/\sqrt{10}, -1/\sqrt{10}, 0]^T$

Example: Computing the SVD

- Since there are no zero singular values, the thin and compressed SVDs are the same and are obtained by $A = U_{3\times2}\Sigma_{2\times2}V_{2\times2}^T$ where

$$\Sigma_{2\times2} = \begin{bmatrix} \sqrt{7} & 0 \\ 0 & \sqrt{5} \end{bmatrix}, \quad V_{2\times2} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}, \quad U_{3\times2} = \begin{bmatrix} \frac{1}{\sqrt{14}} & \frac{3}{\sqrt{10}} \\ \frac{3}{\sqrt{14}} & -\frac{1}{\sqrt{10}} \\ \frac{2}{\sqrt{14}} & 0 \end{bmatrix}$$

Example: Computing the SVD

- For the full SVD, we need to find a vector $\mathbf{u}_3$ completing $\mathbf{u}_1$ and $\mathbf{u}_2$ to an orthonormal basis

Example: Computing the SVD

- For the full SVD, we need to find a vector $\mathbf{u}_3$ completing $\mathbf{u}_1$ and $\mathbf{u}_2$ to an orthonormal basis

- Recall that $\mathbf{u}_1 \times \mathbf{u}_2$ is orthogonal to both $\mathbf{u}_1$ and $\mathbf{u}_2$ thus $\mathbf{u}_3 = \sqrt{14}\mathbf{u}_1 \times \sqrt{10}\mathbf{u}_2 = [2, 6, -10]^T$ and normalizing by its norm ($\mathbf{u}_3 = \mathbf{u}_3/||\mathbf{u}_3||_2$) we obtain $\mathbf{u}_3 = [2, 6, -10]^T/\sqrt{140}$

Example: Computing the SVD

- For the full SVD, we need to find a vector $\mathbf{u}_3$ completing $\mathbf{u}_1$ and $\mathbf{u}_2$ to an orthonormal basis

- Recall that $\mathbf{u}_1 \times \mathbf{u}_2$ is orthogonal to both $\mathbf{u}_1$ and $\mathbf{u}_2$ thus $\mathbf{u}_3 = \sqrt{14}\mathbf{u}_1 \times \sqrt{10}\mathbf{u}_2 = [2, 6, -10]^T$ and normalizing by its norm ($\mathbf{u}_3 = \mathbf{u}_3/||\mathbf{u}_3||_2$) we obtain $\mathbf{u}_3 = [2, 6, -10]^T/\sqrt{140}$

- Therefore, the full SVD is formed as $A = U_{3\times3}\Sigma_{3\times2}V_{2\times2}^T$ where

$$\Sigma_{3\times2} = \begin{bmatrix} \sqrt{7} & 0 \\ 0 & \sqrt{5} \\ 0 & 0 \end{bmatrix}, V_{2\times2} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}, U_{3\times3} = \begin{bmatrix} \frac{1}{\sqrt{14}} & \frac{3}{\sqrt{10}} & \frac{2}{\sqrt{140}} \\ \frac{3}{\sqrt{140}} & -\frac{1}{\sqrt{10}} & \frac{6}{\sqrt{140}} \\ \frac{2}{\sqrt{14}} & 0 & -\frac{10}{\sqrt{140}} \end{bmatrix}$$

The SVD and the Inverse

- If $A \in \mathbb{R}^{n \times n}$ and rank$(A) = n$, then A is invertible and there exists a unique **inverse** $B = A^{-1} \in \mathbb{R}^{n \times n}$ such that $BA = AB = I_{n \times n}$

The SVD and the Inverse

- If $A \in \mathbb{R}^{n \times n}$ and rank$(A) = n$, then A is invertible and there exists a unique **inverse** $B = A^{-1} \in \mathbb{R}^{n \times n}$ such that
$$BA = AB = I_{n \times n}$$

- If $A_{n \times n} = U_{n \times n} \Sigma_{n \times n} V_{n \times n}^T$, then
$$A^{-1} = V \Sigma^{-1} U^T$$

since $U_{n \times n}^T U_{n \times n} = I_{n \times n}$, $U^{-1} = U^T$, and similarly $(V^T)^{-1} = V$

The SVD and the Inverse

- If $A \in \mathbb{R}^{n \times n}$ and rank$(A) = n$, then A is invertible and there exists a unique **inverse** $B = A^{-1} \in \mathbb{R}^{n \times n}$ such that $BA = AB = I_{n \times n}$

- If $A_{n \times n} = U_{n \times n} \Sigma_{n \times n} V_{n \times n}^T$, then

$$A^{-1} = V \Sigma^{-1} U^T$$

  since $U_{n \times n}^T U_{n \times n} = I_{n \times n}$, $U^{-1} = U^T$, and similarly $(V^T)^{-1} = V$

- Combining these observations with the identity $(AB)^{-1} = B^{-1} A^{-1}$ we conclude that

$$
\begin{aligned}
A^{-1} &= (U_{n \times n} \Sigma_{n \times n} V_{n \times n}^T)^{-1} \\
&= (V_{n \times n}^T)^{-1} \Sigma^{-1} U^{-1} \\
&= V \Sigma^{-1} U^T
\end{aligned}
$$

The SVD and the Pseudoinverse

- If $A_{m \times n}$ with $m \geq n$ and rank$(A) = n$, then we define the **pseudoinverse** of A as the matrix $B = A^\dagger$ that satisfies

$$BA = I_{n \times n} \tag{9}$$

and AB is symmetric $\tag{10}$

The SVD and the Pseudoinverse

- If $A_{m \times n}$ with $m \geq n$ and rank$(A) = n$, then we define the **pseudoinverse** of A as the matrix $B = A^{\dagger}$ that satisfies

$$BA = I_{n \times n} \tag{9}$$

$$\text{and } AB \text{ is symmetric} \tag{10}$$

- If $m = n$, then the psuedoinverse coincides with the inverse

The SVD and the Pseudoinverse

- To show that the psuedoinverse is uniquely defined we start by multiplying (9) by A from the left to obtain

$$ABA = A \qquad (11)$$

The SVD and the Pseudoinverse

- To show that the psuedoinverse is uniquely defined we start by multiplying (9) by A from the left to obtain

$$ABA = A \tag{11}$$

- Applying the transpose to both sides of (11) results in

$$A^T(AB)^T = A^T \tag{12}$$

The SVD and the Pseudoinverse

- It follows from (10) and (12) that

$$A^T A B = A^T \qquad (13)$$

The SVD and the Pseudoinverse

- It follows from (10) and (12) that

$$A^T A B = A^T \qquad (13)$$

- Since A has full rank the matrix $A^T A$ is invertible and thus (13) implies that the pseudoinverse $B = A^\dagger$ is uniquely defined and can be computed as

$$A^\dagger = B = (A^T A)^{-1} A^T \qquad (14)$$

The SVD and the Pseudoinverse

- We can also express the pseudoinverse in terms of the SVD of A

The SVD and the Pseudoinverse

- We can also express the pseudoinverse in terms of the SVD of A

- If $A = U\Sigma V^T$, then plugging this into (14) yields

$$A^\dagger = V\Sigma^{-1}U^T \qquad (15)$$

The SVD and the Pseudoinverse

- We can also express the pseudoinverse in terms of the SVD of A

- If $A = U\Sigma V^T$, then plugging this into (14) yields

$$A^\dagger = V\Sigma^{-1}U^T \qquad (15)$$

- Note that (15) implies that $AA^\dagger = UU^T$ (and that $A^\dagger A = I$), that is $AA^\dagger$ is the orthogonal projector onto the image of A (if $m = n$ then clearly this orthogonal projector is I)

The SVD and the Pseudoinverse

- It is worthy to note that the computation of the pseudoinverse by (15) is numerically more stable than that in (14)

The SVD and the Pseudoinverse

- It is worthy to note that the computation of the pseudoinverse by (15) is numerically more stable than that in (14)

- In particular, the multiplication $A^T A$ in (14) followed by matrix inversion accumulates significant numerical error

Review Question

Let

$$A = \begin{bmatrix} -3 & -4 \\ 0 & 0 \\ 0 & 0 \\ 0 & 5 \end{bmatrix}$$

- Find the thin SVD of A (use MATLAB to check your answer)

The Least Squares Problem

- Given $A \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$, the **least squares problem** is to find an $\mathbf{x} \in \mathbb{R}^n$ that minimizes $||A\mathbf{x} - \mathbf{b}||_2$

The Least Squares Problem

- Given $A \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$, the **least squares problem** is to find an $\mathbf{x} \in \mathbb{R}^n$ that minimizes $||A\mathbf{x} - \mathbf{b}||_2$

- The minimizing vector $\mathbf{x}$ is called a **least squares solution** of $A\mathbf{x} = \mathbf{b}$

The Normal Equations

- A solution **x** of the least squares problem is a solution of the linear system

$$A^T A \mathbf{x} = A^T \mathbf{b}$$

and the system is nonsingular iff A has linearly independent columns

The Normal Equations

- A solution **x** of the least squares problem is a solution of the linear system

$$A^T A\mathbf{x} = A^T \mathbf{b}$$

  and the system is nonsingular iff A has linearly independent columns

- The linear system $A^T A\mathbf{x} = A^T \mathbf{b}$ is called the **normal equations**

Solving the Least Squares Problem using SVD

- Since $A^T A \mathbf{x} = A^T \mathbf{b}$ we obtain $\mathbf{x} = A^\dagger \mathbf{b}$ where $A^\dagger = (A^T A)^{-1} A^T$ is the pseudoinverse

Solving the Least Squares Problem using SVD

- Since $A^T A \mathbf{x} = A^T \mathbf{b}$ we obtain $\mathbf{x} = A^\dagger \mathbf{b}$ where $A^\dagger = (A^T A)^{-1} A^T$ is the pseudoinverse

- Using the SVD, we can compute the solution set as

$$\mathbf{x} = A^\dagger \mathbf{b}$$
$$= V \Sigma^{-1} U^T \mathbf{b}$$

  and thus the pseudoinverse provides the optimal solution to the least squares problem

Review Question

Let

$$A = \begin{bmatrix} 1 & 3 & 1 \\ 6 & -1 & 2 \\ 9 & -4 & 3 \\ 7 & -2 & 1 \\ 3 & 8 & 4 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ -1 \\ -3 \\ 7 \\ 9 \end{bmatrix}$$

- Using the MATLAB `svd` command, find a vector **x** that minimizes $||A\mathbf{x} - \mathbf{b}||_2$

Fitting a Line in $\mathbb{R}^2$

- Suppose $\{[x_i, y_i]^T\}_{i=1}^N \subseteq \mathbb{R}^2$ and for a line l in the plane define $t_i = t_i(x_i, y_i, l)$ as the vertical distance from a point $[x_i, y_i]^T$ to the line

Fitting a Line in $\mathbb{R}^2$

- Suppose $\{[x_i, y_i]^T\}_{i=1}^N \subseteq \mathbb{R}^2$ and for a line l in the plane define $t_i = t_i(x_i, y_i, l)$ as the vertical distance from a point $[x_i, y_i]^T$ to the line

- The goal is to minimize $\sum_{i=1}^N t_i^2$ among all possible lines l and if l is $f(x) = ax + b$ we want to minimize $\sum_{i=1}^N (f(x_i) - y_i)^2$

Fitting a Line in $\mathbb{R}^2$

- Suppose $\{[x_i, y_i]^T\}_{i=1}^N \subseteq \mathbb{R}^2$ and for a line l in the plane define $t_i = t_i(x_i, y_i, l)$ as the vertical distance from a point $[x_i, y_i]^T$ to the line

- The goal is to minimize $\sum_{i=1}^N t_i^2$ among all possible lines l and if l is $f(x) = ax + b$ we want to minimize $\sum_{i=1}^N (f(x_i) - y_i)^2$

- Define the A, $\mathbf{x}$, and $\mathbf{b}$ by

$$A = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} a \\ b \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

and minimize $||A\mathbf{x} - \mathbf{b}||_2^2$ over all $\mathbf{x} = [a, b]^T \in \mathbb{R}^2$

Fitting a Line in $\mathbb{R}^{d+1}$

- In the more general case, let $\mathbf{x}_i = [x_{i,1}, \ldots, x_{i,d}, y_i]^T \in \mathbb{R}^{d+1}$

Fitting a Line in $\mathbb{R}^{d+1}$

- In the more general case, let $\mathbf{x}_i = [x_{i,1}, \ldots, x_{i,d}, y_i]^T \in \mathbb{R}^{d+1}$

- We want to fit a linear function
  $f(x_i) = a_1 x_{i,1} + \cdots + a_d x_{i,d} + b$ which minimizes
  $\sum (f(x_i) - y_i)^2$

Fitting a Line in $\mathbb{R}^{d+1}$

- In the more general case, let $\mathbf{x}_i = [x_{i,1}, \ldots, x_{i,d}, y_i]^T \in \mathbb{R}^{d+1}$

- We want to fit a linear function
  $f(x_i) = a_1 x_{i,1} + \cdots + a_d x_{i,d} + b$ which minimizes
  $\sum (f(x_i) - y_i)^2$

- Similar to the $\mathbb{R}^2$ case, define the A, $\mathbf{x}$, and $\mathbf{b}$ by

$$
A = \begin{bmatrix} x_{1,1} & \cdots & x_{1,d} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_{n,1} & \cdots & x_{n,d} & 1 \end{bmatrix}, \quad
\mathbf{x} = \begin{bmatrix} a_1 \\ \vdots \\ a_d \\ b \end{bmatrix}, \quad
\mathbf{b} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}
$$

and minimize $||A\mathbf{x} - \mathbf{b}||_2^2$ over all $\mathbf{x} \in \mathbb{R}^{d+1}$

Fitting a Quadratic Function

- If we need to fit a quadratic function rather than a linear function we can still use the same techniques

Fitting a Quadratic Function

- If we need to fit a quadratic function rather than a linear function we can still use the same techniques

- Suppose $x_i, y_i \in \mathbb{R}$ and we want to fit a function $f(x_i) = ax_i^2 + bx_i + c$ that minimizes the sum of the squares of the vertical distances between the data points and $f(x)$

Fitting a Quadratic Function

- If we need to fit a quadratic function rather than a linear function we can still use the same techniques

- Suppose $x_i, y_i \in \mathbb{R}$ and we want to fit a function $f(x_i) = ax_i^2 + bx_i + c$ that minimizes the sum of the squares of the vertical distances between the data points and $f(x)$

- Define A, $\mathbf{x}$, and $\mathbf{b}$ by

$$A = \begin{bmatrix} x_1^2 & x_1 & 1 \\ \vdots & \vdots & \vdots \\ x_n^2 & x_n & 1 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

and minimize $||A\mathbf{x} - \mathbf{b}||_2^2$ over all $\mathbf{x} \in \mathbb{R}^3$

Principal Component Analysis (PCA)

- In **principal component analysis** (PCA) we find the directions in the data with the most variation, i.e. the eigenvectors corresponding to the largest eigenvalues of the covariance matrix, and project the data onto these directions

Principal Component Analysis (PCA)

- In **principal component analysis** (PCA) we find the directions in the data with the most variation, i.e. the eigenvectors corresponding to the largest eigenvalues of the covariance matrix, and project the data onto these directions

- The motivation for doing this is that the most second order information is located along these directions

Finding the Best d-Dimensional Subspace

- Given the data points $\{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^D$ we want to find l, a d-dimensional subspace ($1 \leq d < D$) which minimizes $\sum_{i=1}^n \text{dist}(\mathbf{x}_i, l)$

Finding the Best d-Dimensional Subspace

- Given the data points $\{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^D$ we want to find l, a d-dimensional subspace ($1 \leq d < D$) which minimizes $\sum_{i=1}^n \text{dist}(\mathbf{x}_i, l)$

- This is the orthogonal distance from the data points to the subspace among all d-dimensional subspaces l of $\mathbb{R}^d$

PCA via SVD

- The **PCA algorithm** for finding this d-dimensional subspace is as follows

PCA via SVD

- The **PCA algorithm** for finding this d-dimensional subspace is as follows

- Step 1: Calculate

$$\mathbf{x}_{cm_i} = \frac{\sum_{j=1}^{n} x_{j,i}}{n}$$

For example, to find the first coordinate of the center of mass $\mathbf{x}_{cm_i}$, sum the first coordinate of all data points and divide by n

PCA via SVD

- The **PCA algorithm** for finding this d-dimensional subspace is as follows

- Step 1: Calculate

$$\mathbf{x}_{cm_i} = \frac{\sum_{j=1}^{n} x_{j,i}}{n}$$

  For example, to find the first coordinate of the center of mass $\mathbf{x}_{cm_i}$, sum the first coordinate of all data points and divide by n

- Step 2: Shift the data points so that the center of mass is at the origin, e.g. $\mathbf{x}_i = \mathbf{x}_i - \mathbf{x}_{cm}$

PCA via SVD

- Step 3: Form

$$A = \begin{bmatrix} \mathbf{x}_i^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix}$$

where the $\mathbf{x}_i$ are the new $\mathbf{x}_i$ formed in step 2

PCA via SVD

- Step 3: Form

$$A = \begin{bmatrix} \mathbf{x}_i^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix}$$

  where the $\mathbf{x}_i$ are the new $\mathbf{x}_i$ formed in step 2

- Step 4: Calculate the right singular vectors $\mathbf{v}_1, \ldots, \mathbf{v}_d$ corresponding to the d largest singular values of A where the SVD of A is $A = U \Sigma V^T$ and the $\mathbf{v}_i$ are the columns of V (or the rows of V^T)

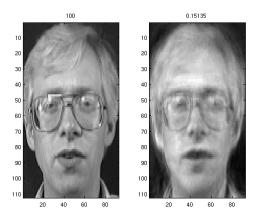# PCA via SVD

- Step 3: Form

$$A = \begin{bmatrix} \mathbf{x}_i^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix}$$

  where the $\mathbf{x}_i$ are the new $\mathbf{x}_i$ formed in step 2

- Step 4: Calculate the right singular vectors $\mathbf{v}_1, \ldots, \mathbf{v}_d$ corresponding to the d largest singular values of A where the SVD of A is $A = U \Sigma V^T$ and the $\mathbf{v}_i$ are the columns of V (or the rows of V^T)

- Step 5: Output $l = \mathbf{x}_{cm} + \text{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_d\}$

Image Approximation using PCA

- The original image (left) compared to its approximating image using d principal components (right).

Summary

- The SVD allows us to obtain fundamental insights into the structure of a matrix

Summary

- The SVD allows us to obtain fundamental insights into the structure of a matrix

- Within the areas of image and data analysis and computer vision, the SVD has numerous applications