A Corgi dog and a yellow rubber duck are positioned in front of a chalkboard. The chalkboard features a graph titled 'GRADIENT FUNCTION' with two overlapping cones, one orange and one blue. Various mathematical notations and equations are written on the board, including 'Nabla', 'f → min', and several equations involving numbers and variables. A clapperboard is visible on the left side of the frame.

Basic linear algebra recap. SVD. Skeleton.
Gradient. Hessian. Matrix calculus.

Daniil Merkulov

Optimization Methods. MIPT

Basic linear algebra background

Vectors and matrices

We will treat all vectors as column vectors by default. The space of real vectors of length n is denoted by \mathbb{R}^n , while the space of real-valued $m \times n$ matrices is denoted by $\mathbb{R}^{m \times n}$. That's it: ¹

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad x^T = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix} \quad x \in \mathbb{R}^n, x_i \in \mathbb{R} \quad (1)$$

¹A full introduction to applied linear algebra can be found in Introduction to Applied Linear Algebra – Vectors, Matrices, and Least Squares - book by Stephen Boyd & Lieven Vandenberghe, which is indicated in the source. Also, a useful refresher for linear algebra is in Appendix A of the book Numerical Optimization by Jorge Nocedal Stephen J. Wright.

Vectors and matrices

We will treat all vectors as column vectors by default. The space of real vectors of length n is denoted by \mathbb{R}^n , while the space of real-valued $m \times n$ matrices is denoted by $\mathbb{R}^{m \times n}$. That's it: ¹

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad x^T = [x_1 \quad x_2 \quad \dots \quad x_n] \quad x \in \mathbb{R}^n, x_i \in \mathbb{R} \quad (1)$$

Similarly, if $A \in \mathbb{R}^{m \times n}$ we denote transposition as $A^T \in \mathbb{R}^{n \times m}$:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \quad A^T = \begin{bmatrix} a_{11} & a_{21} & \dots & a_{m1} \\ a_{12} & a_{22} & \dots & a_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \dots & a_{mn} \end{bmatrix} \quad A \in \mathbb{R}^{m \times n}, a_{ij} \in \mathbb{R}$$

We will write $x \geq 0$ and $x \neq 0$ to indicate componentwise relationships

¹A full introduction to applied linear algebra can be found in Introduction to Applied Linear Algebra – Vectors, Matrices, and Least Squares - book by Stephen Boyd & Lieven Vandenbergh, which is indicated in the source. Also, a useful refresher for linear algebra is in Appendix A of the book Numerical Optimization by Jorge Nocedal Stephen J. Wright.



Figure 1: Equivalent representations of a vector

A matrix is symmetric if $A = A^T$. It is denoted as $A \in \mathbb{S}^n$ (set of square symmetric matrices of dimension n). Note, that only a square matrix could be symmetric by definition.

A matrix is symmetric if $A = A^T$. It is denoted as $A \in \mathbb{S}^n$ (set of square symmetric matrices of dimension n). Note, that only a square matrix could be symmetric by definition.

A matrix $A \in \mathbb{S}^n$ is called **positive (negative) definite** if for all $x \neq 0 : x^T A x > (<) 0$. We denote this as $A \succ (<) 0$. The set of such matrices is denoted as $\mathbb{S}_{++}^n (\mathbb{S}_{--}^n)$

A matrix is symmetric if $A = A^T$. It is denoted as $A \in \mathbb{S}^n$ (set of square symmetric matrices of dimension n). Note, that only a square matrix could be symmetric by definition.

A matrix $A \in \mathbb{S}^n$ is called **positive (negative) definite** if for all $x \neq 0 : x^T A x > (<) 0$. We denote this as $A \succ (<) 0$. The set of such matrices is denoted as $\mathbb{S}_{++}^n (\mathbb{S}_{--}^n)$

A matrix $A \in \mathbb{S}^n$ is called **positive (negative) semidefinite** if for all $x : x^T A x \geq (\leq) 0$. We denote this as $A \succeq (\preceq) 0$. The set of such matrices is denoted as $\mathbb{S}_+^n (\mathbb{S}_-^n)$

Question

Is it correct, that a positive definite matrix has all positive entries?

A matrix is symmetric if $A = A^T$. It is denoted as $A \in \mathbb{S}^n$ (set of square symmetric matrices of dimension n). Note, that only a square matrix could be symmetric by definition.

A matrix $A \in \mathbb{S}^n$ is called **positive (negative) definite** if for all $x \neq 0 : x^T A x > (<) 0$. We denote this as $A \succ (<) 0$. The set of such matrices is denoted as $\mathbb{S}_{++}^n (\mathbb{S}_{--}^n)$

A matrix $A \in \mathbb{S}^n$ is called **positive (negative) semidefinite** if for all $x : x^T A x \geq (\leq) 0$. We denote this as $A \succeq (\preceq) 0$. The set of such matrices is denoted as $\mathbb{S}_+^n (\mathbb{S}_-^n)$

Question

Is it correct, that a positive definite matrix has all positive entries?

Question

Is it correct, that if a matrix is symmetric it should be positive definite?

A matrix is symmetric if $A = A^T$. It is denoted as $A \in \mathbb{S}^n$ (set of square symmetric matrices of dimension n). Note, that only a square matrix could be symmetric by definition.

A matrix $A \in \mathbb{S}^n$ is called **positive (negative) definite** if for all $x \neq 0 : x^T A x > (<) 0$. We denote this as $A \succ (<) 0$. The set of such matrices is denoted as $\mathbb{S}_{++}^n (\mathbb{S}_{--}^n)$

A matrix $A \in \mathbb{S}^n$ is called **positive (negative) semidefinite** if for all $x : x^T A x \geq (\leq) 0$. We denote this as $A \succeq (\preceq) 0$. The set of such matrices is denoted as $\mathbb{S}_+^n (\mathbb{S}_-^n)$

Question

Is it correct, that a positive definite matrix has all positive entries?

Question

Is it correct, that if a matrix is symmetric it should be positive definite?

Question

Is it correct, that if a matrix is positive definite it should be symmetric?

Matrix product (matmul)

Let A be a matrix of size $m \times n$, and B be a matrix of size $n \times p$, and let the product AB be:

$$C = AB$$

then C is a $m \times p$ matrix, with element (i, j) given by:

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}.$$

This operation in a naive form requires $\mathcal{O}(n^3)$ arithmetical operations, where n is usually assumed as the largest dimension of matrices.

Matrix product (matmul)

Let A be a matrix of size $m \times n$, and B be a matrix of size $n \times p$, and let the product AB be:

$$C = AB$$

then C is a $m \times p$ matrix, with element (i, j) given by:

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}.$$

This operation in a naive form requires $\mathcal{O}(n^3)$ arithmetical operations, where n is usually assumed as the largest dimension of matrices.

Question

Is it possible to multiply two matrices faster, than $\mathcal{O}(n^3)$? How about $\mathcal{O}(n^2)$, $\mathcal{O}(n)$?

Matrix by vector product (matvec)

Let A be a matrix of shape $m \times n$, and x be $n \times 1$ vector, then the i -th component of the product:

$$z = Ax$$

is given by:

$$z_i = \sum_{k=1}^n a_{ik} x_k$$

This operation in a naive form requires $\mathcal{O}(n^2)$ arithmetical operations, where n is usually assumed as the largest dimension of matrices.

Remember, that:

- $C = AB \quad C^T = B^T A^T$

Matrix by vector product (matvec)

Let A be a matrix of shape $m \times n$, and x be $n \times 1$ vector, then the i -th component of the product:

$$z = Ax$$

is given by:

$$z_i = \sum_{k=1}^n a_{ik} x_k$$

This operation in a naive form requires $\mathcal{O}(n^2)$ arithmetical operations, where n is usually assumed as the largest dimension of matrices.

Remember, that:

- $C = AB \quad C^T = B^T A^T$
- $AB \neq BA$

Matrix by vector product (matvec)

Let A be a matrix of shape $m \times n$, and x be $n \times 1$ vector, then the i -th component of the product:

$$z = Ax$$

is given by:

$$z_i = \sum_{k=1}^n a_{ik} x_k$$

This operation in a naive form requires $\mathcal{O}(n^2)$ arithmetical operations, where n is usually assumed as the largest dimension of matrices.

Remember, that:

- $C = AB \quad C^T = B^T A^T$
- $AB \neq BA$
- $e^A = \sum_{k=0}^{\infty} \frac{1}{k!} A^k$

Matrix by vector product (matvec)

Let A be a matrix of shape $m \times n$, and x be $n \times 1$ vector, then the i -th component of the product:

$$z = Ax$$

is given by:

$$z_i = \sum_{k=1}^n a_{ik} x_k$$

This operation in a naive form requires $\mathcal{O}(n^2)$ arithmetical operations, where n is usually assumed as the largest dimension of matrices.

Remember, that:

- $C = AB \quad C^T = B^T A^T$
- $AB \neq BA$
- $e^A = \sum_{k=0}^{\infty} \frac{1}{k!} A^k$
- $e^{A+B} \neq e^A e^B$ (but if A and B are commuting matrices, which means that $AB = BA$, $e^{A+B} = e^A e^B$)

Matrix by vector product (matvec)

Let A be a matrix of shape $m \times n$, and x be $n \times 1$ vector, then the i -th component of the product:

$$z = Ax$$

is given by:

$$z_i = \sum_{k=1}^n a_{ik} x_k$$

This operation in a naive form requires $\mathcal{O}(n^2)$ arithmetical operations, where n is usually assumed as the largest dimension of matrices.

Remember, that:

- $C = AB \quad C^T = B^T A^T$
- $AB \neq BA$
- $e^A = \sum_{k=0}^{\infty} \frac{1}{k!} A^k$
- $e^{A+B} \neq e^A e^B$ (but if A and B are commuting matrices, which means that $AB = BA$, $e^{A+B} = e^A e^B$)
- $\langle x, Ay \rangle = \langle A^T x, y \rangle$

Example. Simple yet important idea on matrix computations.

Suppose, you have the following expression

$$b = A_1 A_2 A_3 x,$$

where the $A_1, A_2, A_3 \in \mathbb{R}^{3 \times 3}$ - random square dense matrices and $x \in \mathbb{R}^n$ - vector. You need to compute b .

Which one way is the best to do it?

1. $A_1 A_2 A_3 x$ (from left to right)

Check the simple code snippet after all.

Example. Simple yet important idea on matrix computations.

Suppose, you have the following expression

$$b = A_1 A_2 A_3 x,$$

where the $A_1, A_2, A_3 \in \mathbb{R}^{3 \times 3}$ - random square dense matrices and $x \in \mathbb{R}^n$ - vector. You need to compute b .

Which one way is the best to do it?

1. $A_1 A_2 A_3 x$ (from left to right)
2. $(A_1 (A_2 (A_3 x)))$ (from right to left)

Check the simple  code snippet after all.

Example. Simple yet important idea on matrix computations.

Suppose, you have the following expression

$$b = A_1 A_2 A_3 x,$$

where the $A_1, A_2, A_3 \in \mathbb{R}^{3 \times 3}$ - random square dense matrices and $x \in \mathbb{R}^n$ - vector. You need to compute b .

Which one way is the best to do it?

1. $A_1 A_2 A_3 x$ (from left to right)
2. $(A_1 (A_2 (A_3 x)))$ (from right to left)
3. It does not matter

Check the simple  code snippet after all.

Example. Simple yet important idea on matrix computations.

Suppose, you have the following expression

$$b = A_1 A_2 A_3 x,$$

where the $A_1, A_2, A_3 \in \mathbb{R}^{3 \times 3}$ - random square dense matrices and $x \in \mathbb{R}^n$ - vector. You need to compute b .

Which one way is the best to do it?

1. $A_1 A_2 A_3 x$ (from left to right)
2. $(A_1 (A_2 (A_3 x)))$ (from right to left)
3. It does not matter
4. The results of the first two options will not be the same.

Check the simple  code snippet after all.

Norms

Norm is a **qualitative measure of the smallness of a vector** and is typically denoted as $\|x\|$.

The norm should satisfy certain properties:

1. $\|\alpha x\| = |\alpha| \|x\|, \alpha \in \mathbb{R}$

Norms

Norm is a **qualitative measure of the smallness of a vector** and is typically denoted as $\|x\|$.

The norm should satisfy certain properties:

1. $\|\alpha x\| = |\alpha| \|x\|$, $\alpha \in \mathbb{R}$
2. $\|x + y\| \leq \|x\| + \|y\|$ (triangle inequality)

Norms

Norm is a **qualitative measure of the smallness of a vector** and is typically denoted as $\|x\|$.

The norm should satisfy certain properties:

1. $\|\alpha x\| = |\alpha| \|x\|$, $\alpha \in \mathbb{R}$
2. $\|x + y\| \leq \|x\| + \|y\|$ (triangle inequality)
3. If $\|x\| = 0$ then $x = 0$

Norms

Norm is a **qualitative measure of the smallness of a vector** and is typically denoted as $\|x\|$.

The norm should satisfy certain properties:

1. $\|\alpha x\| = |\alpha| \|x\|$, $\alpha \in \mathbb{R}$
2. $\|x + y\| \leq \|x\| + \|y\|$ (triangle inequality)
3. If $\|x\| = 0$ then $x = 0$

Norms

Norm is a **qualitative measure of the smallness of a vector** and is typically denoted as $\|x\|$.

The norm should satisfy certain properties:

1. $\|\alpha x\| = |\alpha| \|x\|$, $\alpha \in \mathbb{R}$
2. $\|x + y\| \leq \|x\| + \|y\|$ (triangle inequality)
3. If $\|x\| = 0$ then $x = 0$

The distance between two vectors is then defined as

$$d(x, y) = \|x - y\|.$$

The most well-known and widely used norm is **Euclidean norm**:

$$\|x\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2},$$

which corresponds to the distance in our real life. If the vectors have complex elements, we use their modulus. Euclidean norm, or 2-norm, is a subclass of an important class of p -norms:

$$\|x\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{1/p}.$$

p -norm of a vector

There are two very important special cases. The infinity norm, or Chebyshev norm is defined as the element of the maximal absolute value:

$$\|x\|_{\infty} = \max_i |x_i|$$

p -norm of a vector

There are two very important special cases. The infinity norm, or Chebyshev norm is defined as the element of the maximal absolute value:

$$\|x\|_{\infty} = \max_i |x_i|$$

L_1 norm (or **Manhattan distance**) which is defined as the sum of modules of the elements of x :

$$\|x\|_1 = \sum_i |x_i|$$

p -norm of a vector

There are two very important special cases. The infinity norm, or Chebyshev norm is defined as the element of the maximal absolute value:

$$\|x\|_{\infty} = \max_i |x_i|$$

L_1 norm (or **Manhattan distance**) which is defined as the sum of modules of the elements of x :

$$\|x\|_1 = \sum_i |x_i|$$

L_1 norm plays a very important role: it all relates to the **compressed sensing** methods that emerged in the mid-00s as one of the most popular research topics. The code for the picture below is available *here*. Check also *this* video.

Unit disk in the p -th norm

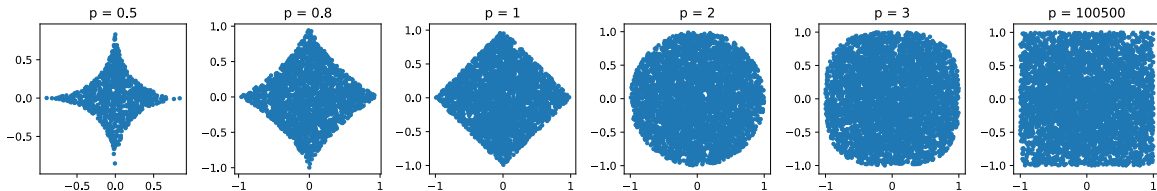


Figure 2: Balls in different norms on a plane

Matrix norms

In some sense there is no big difference between matrices and vectors (you can vectorize the matrix), and here comes the simplest matrix norm **Frobenius** norm:

$$\|A\|_F = \left(\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2 \right)^{1/2}$$

Matrix norms

In some sense there is no big difference between matrices and vectors (you can vectorize the matrix), and here comes the simplest matrix norm **Frobenius** norm:

$$\|A\|_F = \left(\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2 \right)^{1/2}$$

Spectral norm, $\|A\|_2$ is one of the most used matrix norms (along with the Frobenius norm).

$$\|A\|_2 = \sup_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2},$$

It can not be computed directly from the entries using a simple formula, like the Frobenius norm, however, there are efficient algorithms to compute it. It is directly related to the **singular value decomposition** (SVD) of the matrix. It holds

$$\|A\|_2 = \sigma_1(A) = \sqrt{\lambda_{\max}(A^T A)}$$

where $\sigma_1(A)$ is the largest singular value of the matrix A .

Scalar product

The standard **scalar (inner) product** between vectors x and y from \mathbb{R}^n is given by

$$\langle x, y \rangle = x^T y = \sum_{i=1}^n x_i y_i = y^T x = \langle y, x \rangle$$

Here x_i and y_i are the scalar i -th components of corresponding vectors.

Example

Prove, that you can switch the position of a matrix inside a scalar product with transposition: $\langle x, Ay \rangle = \langle A^T x, y \rangle$ and $\langle x, yB \rangle = \langle xB^T, y \rangle$

Matrix scalar product

The standard **scalar (inner) product** between matrices X and Y from $\mathbb{R}^{m \times n}$ is given by

$$\langle X, Y \rangle = \text{tr}(X^T Y) = \sum_{i=1}^m \sum_{j=1}^n X_{ij} Y_{ij} = \text{tr}(Y^T X) = \langle Y, X \rangle$$

Question

Is there any connection between the Frobenious norm $\| \cdot \|_F$ and scalar product between matrices $\langle \cdot, \cdot \rangle$?

Eigenvectors and eigenvalues

A scalar value λ is an eigenvalue of the $n \times n$ matrix A if there is a nonzero vector q such that

$$Aq = \lambda q.$$

The vector q is called an eigenvector of A . The matrix A is nonsingular if none of its eigenvalues are zero. The eigenvalues of symmetric matrices are all real numbers, while nonsymmetric matrices may have imaginary eigenvalues. If the matrix is positive definite as well as symmetric, its eigenvalues are all positive real numbers.

Eigenvectors and eigenvalues

i Theorem

$$A \succeq (\succ) 0 \Leftrightarrow \text{all eigenvalues of } A \text{ are } \geq (>) 0$$

i Proof

1. \rightarrow Suppose some eigenvalue λ is negative and let x denote its corresponding eigenvector. Then

$$Ax = \lambda x \rightarrow x^T Ax = \lambda x^T x < 0$$

which contradicts the condition of $A \succeq 0$.

Eigenvectors and eigenvalues

i Theorem

$$A \succeq (\succ) 0 \Leftrightarrow \text{all eigenvalues of } A \text{ are } \geq (>) 0$$

i Proof

1. \rightarrow Suppose some eigenvalue λ is negative and let x denote its corresponding eigenvector. Then

$$Ax = \lambda x \rightarrow x^T Ax = \lambda x^T x < 0$$

which contradicts the condition of $A \succeq 0$.

2. \leftarrow For any symmetric matrix, we can pick a set of eigenvectors v_1, \dots, v_n that form an orthogonal basis of \mathbb{R}^n . Pick any $x \in \mathbb{R}^n$.

$$\begin{aligned} x^T Ax &= (\alpha_1 v_1 + \dots + \alpha_n v_n)^T A(\alpha_1 v_1 + \dots + \alpha_n v_n) \\ &= \sum \alpha_i^2 v_i^T A v_i = \sum \alpha_i^2 \lambda_i v_i^T v_i \geq 0 \end{aligned}$$

here we have used the fact that $v_i^T v_j = 0$, for $i \neq j$.

Eigendecomposition (spectral decomposition)

Suppose $A \in S_n$, i.e., A is a real symmetric $n \times n$ matrix. Then A can be factorized as

$$A = Q\Lambda Q^T,$$

²A good cheat sheet with matrix decomposition is available at the NLA course website.

Eigendecomposition (spectral decomposition)

Suppose $A \in S_n$, i.e., A is a real symmetric $n \times n$ matrix. Then A can be factorized as

$$A = Q\Lambda Q^T,$$

where $Q \in \mathbb{R}^{n \times n}$ is orthogonal, i.e., satisfies $Q^T Q = I$, and $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$. The (real) numbers λ_i are the eigenvalues of A and are the roots of the characteristic polynomial $\det(A - \lambda I)$. The columns of Q form an orthonormal set of eigenvectors of A . The factorization is called the spectral decomposition or (symmetric) eigenvalue decomposition of A .²

²A good cheat sheet with matrix decomposition is available at the NLA course website.

Eigendecomposition (spectral decomposition)

Suppose $A \in S_n$, i.e., A is a real symmetric $n \times n$ matrix. Then A can be factorized as

$$A = Q\Lambda Q^T,$$

where $Q \in \mathbb{R}^{n \times n}$ is orthogonal, i.e., satisfies $Q^T Q = I$, and $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$. The (real) numbers λ_i are the eigenvalues of A and are the roots of the characteristic polynomial $\det(A - \lambda I)$. The columns of Q form an orthonormal set of eigenvectors of A . The factorization is called the spectral decomposition or (symmetric) eigenvalue decomposition of A .²

We usually order the eigenvalues as $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$. We use the notation $\lambda_i(A)$ to refer to the i -th largest eigenvalue of $A \in S$. We usually write the largest or maximum eigenvalue as $\lambda_1(A) = \lambda_{\max}(A)$, and the least or minimum eigenvalue as $\lambda_n(A) = \lambda_{\min}(A)$.

²A good cheat sheet with matrix decomposition is available at the NLA course website.

Eigenvalues

The largest and smallest eigenvalues satisfy

$$\lambda_{\min}(A) = \inf_{x \neq 0} \frac{x^T A x}{x^T x}, \quad \lambda_{\max}(A) = \sup_{x \neq 0} \frac{x^T A x}{x^T x}$$

Eigenvalues

The largest and smallest eigenvalues satisfy

$$\lambda_{\min}(A) = \inf_{x \neq 0} \frac{x^T A x}{x^T x}, \quad \lambda_{\max}(A) = \sup_{x \neq 0} \frac{x^T A x}{x^T x}$$

and consequently $\forall x \in \mathbb{R}^n$ (Rayleigh quotient):

$$\lambda_{\min}(A) x^T x \leq x^T A x \leq \lambda_{\max}(A) x^T x$$

Eigenvalues

The largest and smallest eigenvalues satisfy

$$\lambda_{\min}(A) = \inf_{x \neq 0} \frac{x^T A x}{x^T x}, \quad \lambda_{\max}(A) = \sup_{x \neq 0} \frac{x^T A x}{x^T x}$$

and consequently $\forall x \in \mathbb{R}^n$ (Rayleigh quotient):

$$\lambda_{\min}(A) x^T x \leq x^T A x \leq \lambda_{\max}(A) x^T x$$

The **condition number** of a nonsingular matrix is defined as

$$\kappa(A) = \|A\| \|A^{-1}\|$$

Eigenvalues

The largest and smallest eigenvalues satisfy

$$\lambda_{\min}(A) = \inf_{x \neq 0} \frac{x^T A x}{x^T x}, \quad \lambda_{\max}(A) = \sup_{x \neq 0} \frac{x^T A x}{x^T x}$$

and consequently $\forall x \in \mathbb{R}^n$ (Rayleigh quotient):

$$\lambda_{\min}(A) x^T x \leq x^T A x \leq \lambda_{\max}(A) x^T x$$

The **condition number** of a nonsingular matrix is defined as

$$\kappa(A) = \|A\| \|A^{-1}\|$$

If we use spectral matrix norm, we can get:

$$\kappa(A) = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}$$

If, moreover, $A \in \mathbb{S}_{++}^n$: $\kappa(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}$

Singular value decomposition

Suppose $A \in \mathbb{R}^{m \times n}$ with $\text{rank } A = r$. Then A can be factored as

$$A = U\Sigma V^T$$

Singular value decomposition

Suppose $A \in \mathbb{R}^{m \times n}$ with $\text{rank } A = r$. Then A can be factored as

$$A = U\Sigma V^T$$

where $U \in \mathbb{R}^{m \times r}$ satisfies $U^T U = I$, $V \in \mathbb{R}^{n \times r}$ satisfies $V^T V = I$, and Σ is a diagonal matrix with $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_r)$, such that

Singular value decomposition

Suppose $A \in \mathbb{R}^{m \times n}$ with $\text{rank } A = r$. Then A can be factored as

$$A = U\Sigma V^T$$

where $U \in \mathbb{R}^{m \times r}$ satisfies $U^T U = I$, $V \in \mathbb{R}^{n \times r}$ satisfies $V^T V = I$, and Σ is a diagonal matrix with $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_r)$, such that

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0.$$

Singular value decomposition

Suppose $A \in \mathbb{R}^{m \times n}$ with $\text{rank } A = r$. Then A can be factored as

$$A = U\Sigma V^T$$

where $U \in \mathbb{R}^{m \times r}$ satisfies $U^T U = I$, $V \in \mathbb{R}^{n \times r}$ satisfies $V^T V = I$, and Σ is a diagonal matrix with $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_r)$, such that

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0.$$

This factorization is called the **singular value decomposition (SVD)** of A . The columns of U are called left singular vectors of A , the columns of V are right singular vectors, and the numbers σ_i are the singular values. The singular value decomposition can be written as

$$A = \sum_{i=1}^r \sigma_i u_i v_i^T,$$

where $u_i \in \mathbb{R}^m$ are the left singular vectors, and $v_i \in \mathbb{R}^n$ are the right singular vectors.

Singular value decomposition

Question

Suppose, matrix $A \in \mathbb{S}_{++}^n$. What can we say about the connection between its eigenvalues and singular values?

Singular value decomposition

Question

Suppose, matrix $A \in \mathbb{S}_{++}^n$. What can we say about the connection between its eigenvalues and singular values?

Question

How do the singular values of a matrix relate to its eigenvalues, especially for a symmetric matrix?

Skeleton decomposition

Simple, yet very interesting decomposition is Skeleton decomposition, which can be written in two forms:

$$A = UV^T \quad A = \hat{C}\hat{A}^{-1}\hat{R}$$

Skeleton decomposition

Simple, yet very interesting decomposition is Skeleton decomposition, which can be written in two forms:

$$A = UV^T \quad A = \hat{C}\hat{A}^{-1}\hat{R}$$

The latter expression refers to the fun fact: you can randomly choose r linearly independent columns of a matrix and any r linearly independent rows of a matrix and store only them with the ability to reconstruct the whole matrix exactly.

Skeleton decomposition

Simple, yet very interesting decomposition is Skeleton decomposition, which can be written in two forms:

$$A = UV^T \quad A = \hat{C}\hat{A}^{-1}\hat{R}$$

The latter expression refers to the fun fact: you can randomly choose r linearly independent columns of a matrix and any r linearly independent rows of a matrix and store only them with the ability to reconstruct the whole matrix exactly.

Use cases for Skeleton decomposition are:

- Model reduction, data compression, and speedup of computations in numerical analysis: given rank- r matrix with $r \ll n, m$ one needs to store $\mathcal{O}((n+m)r) \ll nm$ elements.



Figure 3: Illustration of Skeleton decomposition

Skeleton decomposition

Simple, yet very interesting decomposition is Skeleton decomposition, which can be written in two forms:

$$A = UV^T \quad A = \hat{C}\hat{A}^{-1}\hat{R}$$

The latter expression refers to the fun fact: you can randomly choose r linearly independent columns of a matrix and any r linearly independent rows of a matrix and store only them with the ability to reconstruct the whole matrix exactly.

Use cases for Skeleton decomposition are:

- Model reduction, data compression, and speedup of computations in numerical analysis: given rank- r matrix with $r \ll n, m$ one needs to store $\mathcal{O}((n+m)r) \ll nm$ elements.
- Feature extraction in machine learning, where it is also known as matrix factorization

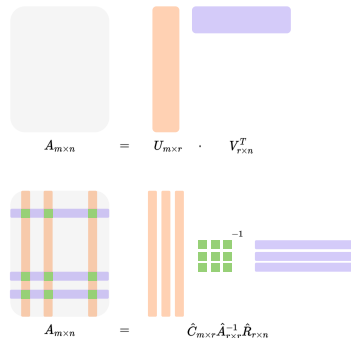


Figure 3: Illustration of Skeleton decomposition

Skeleton decomposition

Simple, yet very interesting decomposition is Skeleton decomposition, which can be written in two forms:

$$A = UV^T \quad A = \hat{C}\hat{A}^{-1}\hat{R}$$

The latter expression refers to the fun fact: you can randomly choose r linearly independent columns of a matrix and any r linearly independent rows of a matrix and store only them with the ability to reconstruct the whole matrix exactly.

Use cases for Skeleton decomposition are:

- Model reduction, data compression, and speedup of computations in numerical analysis: given rank- r matrix with $r \ll n, m$ one needs to store $\mathcal{O}((n+m)r) \ll nm$ elements.
- Feature extraction in machine learning, where it is also known as matrix factorization
- All applications where SVD applies, since Skeleton decomposition can be transformed into truncated SVD form.

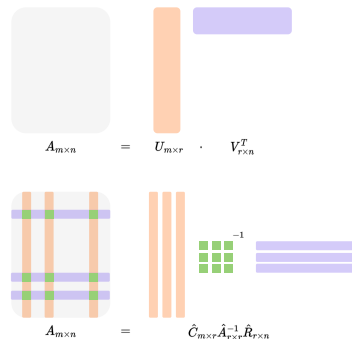


Figure 3: Illustration of Skeleton decomposition

Canonical tensor decomposition

One can consider the generalization of Skeleton decomposition to the higher order data structure, like tensors, which implies representing the tensor as a sum of r primitive tensors.



Figure 4: Illustration of Canonical Polyadic decomposition

i Example

Note, that there are many tensor decompositions: Canonical, Tucker, Tensor Train (TT), Tensor Ring (TR), and others. In the tensor case, we do not have a straightforward definition of *rank* for all types of decompositions. For example, for TT decomposition rank is not a scalar, but a vector.

Determinant and trace

The determinant and trace can be expressed in terms of the eigenvalues

$$\det A = \prod_{i=1}^n \lambda_i, \quad \operatorname{tr} A = \sum_{i=1}^n \lambda_i$$

The determinant has several appealing (and revealing) properties. For instance,

- $\det A = 0$ if and only if A is singular;

Determinant and trace

The determinant and trace can be expressed in terms of the eigenvalues

$$\det A = \prod_{i=1}^n \lambda_i, \quad \operatorname{tr} A = \sum_{i=1}^n \lambda_i$$

The determinant has several appealing (and revealing) properties. For instance,

- $\det A = 0$ if and only if A is singular;
- $\det AB = (\det A)(\det B)$;

Determinant and trace

The determinant and trace can be expressed in terms of the eigenvalues

$$\det A = \prod_{i=1}^n \lambda_i, \quad \operatorname{tr} A = \sum_{i=1}^n \lambda_i$$

The determinant has several appealing (and revealing) properties. For instance,

- $\det A = 0$ if and only if A is singular;
- $\det AB = (\det A)(\det B)$;
- $\det A^{-1} = \frac{1}{\det A}$.

Determinant and trace

The determinant and trace can be expressed in terms of the eigenvalues

$$\det A = \prod_{i=1}^n \lambda_i, \quad \operatorname{tr} A = \sum_{i=1}^n \lambda_i$$

The determinant has several appealing (and revealing) properties. For instance,

- $\det A = 0$ if and only if A is singular;
- $\det AB = (\det A)(\det B)$;
- $\det A^{-1} = \frac{1}{\det A}$.

Determinant and trace

The determinant and trace can be expressed in terms of the eigenvalues

$$\det A = \prod_{i=1}^n \lambda_i, \quad \operatorname{tr} A = \sum_{i=1}^n \lambda_i$$

The determinant has several appealing (and revealing) properties. For instance,

- $\det A = 0$ if and only if A is singular;
- $\det AB = (\det A)(\det B)$;
- $\det A^{-1} = \frac{1}{\det A}$.

Don't forget about the cyclic property of a trace for arbitrary matrices A, B, C, D (assuming, that all dimensions are consistent):

$$\operatorname{tr}(ABCD) = \operatorname{tr}(DABC) = \operatorname{tr}(CDAB) = \operatorname{tr}(BCDA)$$

Determinant and trace

The determinant and trace can be expressed in terms of the eigenvalues

$$\det A = \prod_{i=1}^n \lambda_i, \quad \operatorname{tr} A = \sum_{i=1}^n \lambda_i$$

The determinant has several appealing (and revealing) properties. For instance,

- $\det A = 0$ if and only if A is singular;
- $\det AB = (\det A)(\det B)$;
- $\det A^{-1} = \frac{1}{\det A}$.

Don't forget about the cyclic property of a trace for arbitrary matrices A, B, C, D (assuming, that all dimensions are consistent):

$$\operatorname{tr}(ABCD) = \operatorname{tr}(DABC) = \operatorname{tr}(CDAB) = \operatorname{tr}(BCDA)$$

Question

How does the determinant of a matrix relate to its invertibility?

First-order Taylor approximation

The first-order Taylor approximation, also known as the linear approximation, is centered around some point x_0 . If $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a differentiable function, then its first-order Taylor approximation is given by:

$$f_{x_0}^I(x) = f(x_0) + \nabla f(x_0)^T (x - x_0)$$

Where:

- $f(x_0)$ is the value of the function at the point x_0 .

First-order Taylor approximation

The first-order Taylor approximation, also known as the linear approximation, is centered around some point x_0 . If $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a differentiable function, then its first-order Taylor approximation is given by:

$$f_{x_0}^I(x) = f(x_0) + \nabla f(x_0)^T(x - x_0)$$

Where:

- $f(x_0)$ is the value of the function at the point x_0 .
- $\nabla f(x_0)$ is the gradient of the function at the point x_0 .

First-order Taylor approximation

The first-order Taylor approximation, also known as the linear approximation, is centered around some point x_0 . If $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a differentiable function, then its first-order Taylor approximation is given by:

$$f_{x_0}^I(x) = f(x_0) + \nabla f(x_0)^T (x - x_0)$$

Where:

- $f(x_0)$ is the value of the function at the point x_0 .
- $\nabla f(x_0)$ is the gradient of the function at the point x_0 .

First-order Taylor approximation

The first-order Taylor approximation, also known as the linear approximation, is centered around some point x_0 . If $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a differentiable function, then its first-order Taylor approximation is given by:

$$f_{x_0}^I(x) = f(x_0) + \nabla f(x_0)^T(x - x_0)$$

Where:

- $f(x_0)$ is the value of the function at the point x_0 .
- $\nabla f(x_0)$ is the gradient of the function at the point x_0 .

It is very usual to replace the $f(x)$ with $f_{x_0}^I(x)$ near the point x_0 for simple analysis of some approaches.



Figure 5: First order Taylor approximation near the point x_0

Second-order Taylor approximation

The second-order Taylor approximation, also known as the quadratic approximation, includes the curvature of the function. For a twice-differentiable function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, its second-order Taylor approximation centered at some point x_0 is:

$$f_{x_0}^{II}(x) = f(x_0) + \nabla f(x_0)^T(x - x_0) + \frac{1}{2}(x - x_0)^T \nabla^2 f(x_0)(x - x_0)$$

Where $\nabla^2 f(x_0)$ is the Hessian matrix of f at the point x_0 .

Second-order Taylor approximation

The second-order Taylor approximation, also known as the quadratic approximation, includes the curvature of the function. For a twice-differentiable function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, its second-order Taylor approximation centered at some point x_0 is:

$$f_{x_0}^{II}(x) = f(x_0) + \nabla f(x_0)^T(x - x_0) + \frac{1}{2}(x - x_0)^T \nabla^2 f(x_0)(x - x_0)$$

Where $\nabla^2 f(x_0)$ is the Hessian matrix of f at the point x_0 .

When using the linear approximation of the function is not sufficient one can consider replacing the $f(x)$ with $f_{x_0}^{II}(x)$ near the point x_0 . In general, Taylor approximations give us a way to locally approximate functions. The first-order approximation is a plane tangent to the function at the point x_0 , while the second-order approximation includes the curvature and is represented by a parabola. These approximations are especially useful in optimization and numerical methods because they provide a tractable way to work with complex functions.

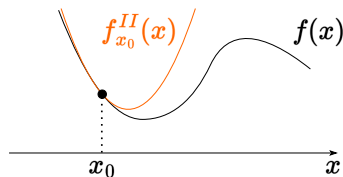


Figure 6: Second order Taylor approximation near the point x_0

Matrix calculus

Gradient

Let $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$, then vector, which contains all first-order partial derivatives:

$$\nabla f(x) = \frac{df}{dx} = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$

Gradient

Let $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$, then vector, which contains all first-order partial derivatives:

$$\nabla f(x) = \frac{df}{dx} = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$

named gradient of $f(x)$. This vector indicates the direction of the steepest ascent. Thus, vector $-\nabla f(x)$ means the direction of the steepest descent of the function in the point. Moreover, the gradient vector is always orthogonal to the contour line in the point.

i Example

For the function $f(x, y) = x^2 + y^2$, the gradient is:

$$\nabla f(x, y) = \begin{bmatrix} 2x \\ 2y \end{bmatrix}$$

This gradient points in the direction of the steepest ascent of the function.

i Question

How does the magnitude of the gradient relate to the steepness of the function?

Hessian

Let $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$, then matrix, containing all the second order partial derivatives:

$$f''(x) = \nabla^2 f(x) = \frac{\partial^2 f}{\partial x_i \partial x_j} = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{pmatrix}$$

Hessian

Let $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$, then matrix, containing all the second order partial derivatives:

$$f''(x) = \nabla^2 f(x) = \frac{\partial^2 f}{\partial x_i \partial x_j} = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{pmatrix}$$

In fact, Hessian could be a tensor in such a way: $(f(x) : \mathbb{R}^n \rightarrow \mathbb{R}^m)$ is just 3d tensor, every slice is just hessian of corresponding scalar function $(\nabla^2 f_1(x), \dots, \nabla^2 f_m(x))$.

i Example

For the function $f(x, y) = x^2 + y^2$, the Hessian is:

$$H_f(x, y) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

This matrix provides information about the curvature of the function in different directions.

i Question

How can the Hessian matrix be used to determine the concavity or convexity of a function?

Schwartz theorem

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a function. If the mixed partial derivatives $\frac{\partial^2 f}{\partial x_i \partial x_j}$ and $\frac{\partial^2 f}{\partial x_j \partial x_i}$ are both continuous on an open set containing a point a , then they are equal at the point a . That is,

$$\frac{\partial^2 f}{\partial x_i \partial x_j}(a) = \frac{\partial^2 f}{\partial x_j \partial x_i}(a)$$

Schwartz theorem

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a function. If the mixed partial derivatives $\frac{\partial^2 f}{\partial x_i \partial x_j}$ and $\frac{\partial^2 f}{\partial x_j \partial x_i}$ are both continuous on an open set containing a point a , then they are equal at the point a . That is,

$$\frac{\partial^2 f}{\partial x_i \partial x_j}(a) = \frac{\partial^2 f}{\partial x_j \partial x_i}(a)$$

Given the Schwartz theorem, if the mixed partials are continuous on an open set, the Hessian matrix is symmetric. That means the entries above the main diagonal mirror those below the main diagonal:

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial^2 f}{\partial x_j \partial x_i} \quad \nabla^2 f(x) = (\nabla^2 f(x))^T$$

This symmetry simplifies computations and analysis involving the Hessian matrix in various applications, particularly in optimization.

i Schwartz counterexample

$$f(x, y) = \begin{cases} \frac{xy(x^2 - y^2)}{x^2 + y^2} & \text{for } (x, y) \neq (0, 0), \\ 0 & \text{for } (x, y) = (0, 0). \end{cases}$$

Counterexample ♣



One can verify, that $\frac{\partial^2 f}{\partial x \partial y}(0,0) \neq \frac{\partial^2 f}{\partial y \partial x}(0,0)$, although the mixed partial derivatives do exist, and at every other point the symmetry does hold.

Jacobian

The extension of the gradient of multidimensional $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is the following matrix:

$$J_f = f'(x) = \frac{df}{dx^T} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}$$

This matrix provides information about the rate of change of the function with respect to its inputs.

i Question

Can we somehow connect those three definitions above (gradient, jacobian, and hessian) using a single correct statement?

i Example

For the function

$$f(x, y) = \begin{bmatrix} x + y \\ x - y \end{bmatrix},$$

the Jacobian is:

$$J_f(x, y) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

i Question

How does the Jacobian matrix relate to the gradient for scalar-valued functions?

Summary

$$f(x) : X \rightarrow Y; \quad \frac{\partial f(x)}{\partial x} \in G$$

X	Y	G	Name
\mathbb{R}	\mathbb{R}	\mathbb{R}	$f'(x)$ (derivative)
\mathbb{R}^n	\mathbb{R}	\mathbb{R}^n	$\frac{\partial f}{\partial x_i}$ (gradient)
\mathbb{R}^n	\mathbb{R}^m	$\mathbb{R}^{n \times m}$	$\frac{\partial f_i}{\partial x_j}$ (jacobian)
$\mathbb{R}^{m \times n}$	\mathbb{R}	$\mathbb{R}^{m \times n}$	$\frac{\partial f}{\partial x_{ij}}$

Theorem

Let $x \in S$ be an interior point of the set S , and let $D : U \rightarrow V$ be a linear operator. We say that the function f is differentiable at the point x with derivative D if for all sufficiently small $h \in U$ the following decomposition holds:

$$f(x + h) = f(x) + D[h] + o(\|h\|)$$

If for any linear operator $D : U \rightarrow V$ the function f is not differentiable at the point x with derivative D , then we say that f is not differentiable at the point x .

Differentials

After obtaining the differential notation of df we can retrieve the gradient using the following formula:

$$df(x) = \langle \nabla f(x), dx \rangle$$

Differentials

After obtaining the differential notation of df we can retrieve the gradient using the following formula:

$$df(x) = \langle \nabla f(x), dx \rangle$$

Then, if we have a differential of the above form and we need to calculate the second derivative of the matrix/vector function, we treat “old” dx as the constant dx_1 , then calculate $d(df) = d^2f(x)$

$$d^2f(x) = \langle \nabla^2 f(x) dx_1, dx \rangle = \langle H_f(x) dx_1, dx \rangle$$

Differential properties

Let A and B be the constant matrices, while X and Y are the variables (or matrix functions).

- $dA = 0$

Differential properties

Let A and B be the constant matrices, while X and Y are the variables (or matrix functions).

- $dA = 0$
- $d(\alpha X) = \alpha(dX)$

Differential properties

Let A and B be the constant matrices, while X and Y are the variables (or matrix functions).

- $dA = 0$
- $d(\alpha X) = \alpha(dX)$
- $d(AXB) = A(dX)B$

Differential properties

Let A and B be the constant matrices, while X and Y are the variables (or matrix functions).

- $dA = 0$
- $d(\alpha X) = \alpha(dX)$
- $d(AXB) = A(dX)B$
- $d(X + Y) = dX + dY$

Differential properties

Let A and B be the constant matrices, while X and Y are the variables (or matrix functions).

- $dA = 0$
- $d(\alpha X) = \alpha(dX)$
- $d(AXB) = A(dX)B$
- $d(X + Y) = dX + dY$
- $d(X^T) = (dX)^T$

Differential properties

Let A and B be the constant matrices, while X and Y are the variables (or matrix functions).

- $dA = 0$
- $d(\alpha X) = \alpha(dX)$
- $d(AXB) = A(dX)B$
- $d(X + Y) = dX + dY$
- $d(X^T) = (dX)^T$
- $d(XY) = (dX)Y + X(dY)$

Differential properties

Let A and B be the constant matrices, while X and Y are the variables (or matrix functions).

- $dA = 0$
- $d(\alpha X) = \alpha(dX)$
- $d(AXB) = A(dX)B$
- $d(X + Y) = dX + dY$
- $d(X^T) = (dX)^T$
- $d(XY) = (dX)Y + X(dY)$
- $d\langle X, Y \rangle = \langle dX, Y \rangle + \langle X, dY \rangle$

Differential properties

Let A and B be the constant matrices, while X and Y are the variables (or matrix functions).

- $dA = 0$
- $d(\alpha X) = \alpha(dX)$
- $d(AXB) = A(dX)B$
- $d(X + Y) = dX + dY$
- $d(X^T) = (dX)^T$
- $d(XY) = (dX)Y + X(dY)$
- $d\langle X, Y \rangle = \langle dX, Y \rangle + \langle X, dY \rangle$

- $$d\left(\frac{X}{\phi}\right) = \frac{\phi dX - (d\phi)X}{\phi^2}$$

Differential properties

Let A and B be the constant matrices, while X and Y are the variables (or matrix functions).

- $dA = 0$
- $d(\alpha X) = \alpha(dX)$
- $d(AXB) = A(dX)B$
- $d(X + Y) = dX + dY$
- $d(X^T) = (dX)^T$
- $d(XY) = (dX)Y + X(dY)$
- $d\langle X, Y \rangle = \langle dX, Y \rangle + \langle X, dY \rangle$

- $d\left(\frac{X}{\phi}\right) = \frac{\phi dX - (d\phi)X}{\phi^2}$
- $d(\det X) = \det X \langle X^{-T}, dX \rangle$

Differential properties

Let A and B be the constant matrices, while X and Y are the variables (or matrix functions).

- $dA = 0$
- $d(\alpha X) = \alpha(dX)$
- $d(AXB) = A(dX)B$
- $d(X + Y) = dX + dY$
- $d(X^T) = (dX)^T$
- $d(XY) = (dX)Y + X(dY)$
- $d\langle X, Y \rangle = \langle dX, Y \rangle + \langle X, dY \rangle$
- $d\left(\frac{X}{\phi}\right) = \frac{\phi dX - (d\phi)X}{\phi^2}$
- $d(\det X) = \det X \langle X^{-T}, dX \rangle$
- $d(\operatorname{tr} X) = \langle I, dX \rangle$

Differential properties

Let A and B be the constant matrices, while X and Y are the variables (or matrix functions).

- $dA = 0$
- $d(\alpha X) = \alpha(dX)$
- $d(AXB) = A(dX)B$
- $d(X + Y) = dX + dY$
- $d(X^T) = (dX)^T$
- $d(XY) = (dX)Y + X(dY)$
- $d\langle X, Y \rangle = \langle dX, Y \rangle + \langle X, dY \rangle$
- $d\left(\frac{X}{\phi}\right) = \frac{\phi dX - (d\phi)X}{\phi^2}$
- $d(\det X) = \det X \langle X^{-T}, dX \rangle$
- $d(\operatorname{tr} X) = \langle I, dX \rangle$
- $df(g(x)) = \frac{df}{dg} \cdot dg(x)$

Differential properties

Let A and B be the constant matrices, while X and Y are the variables (or matrix functions).

- $dA = 0$
- $d(\alpha X) = \alpha(dX)$
- $d(AXB) = A(dX)B$
- $d(X + Y) = dX + dY$
- $d(X^T) = (dX)^T$
- $d(XY) = (dX)Y + X(dY)$
- $d\langle X, Y \rangle = \langle dX, Y \rangle + \langle X, dY \rangle$
- $d\left(\frac{X}{\phi}\right) = \frac{\phi dX - (d\phi)X}{\phi^2}$
- $d(\det X) = \det X \langle X^{-T}, dX \rangle$
- $d(\operatorname{tr} X) = \langle I, dX \rangle$
- $df(g(x)) = \frac{df}{dg} \cdot dg(x)$
- $H = (J(\nabla f))^T$

Differential properties

Let A and B be the constant matrices, while X and Y are the variables (or matrix functions).

- $dA = 0$
- $d(\alpha X) = \alpha(dX)$
- $d(AXB) = A(dX)B$
- $d(X + Y) = dX + dY$
- $d(X^T) = (dX)^T$
- $d(XY) = (dX)Y + X(dY)$
- $d\langle X, Y \rangle = \langle dX, Y \rangle + \langle X, dY \rangle$

- $d\left(\frac{X}{\phi}\right) = \frac{\phi dX - (d\phi)X}{\phi^2}$
- $d(\det X) = \det X \langle X^{-T}, dX \rangle$
- $d(\operatorname{tr} X) = \langle I, dX \rangle$
- $df(g(x)) = \frac{df}{dg} \cdot dg(x)$
- $H = (J(\nabla f))^T$
- $d(X^{-1}) = -X^{-1}(dX)X^{-1}$

Matrix calculus. Example 1

Example

Find $df, \nabla f(x)$, if $f(x) = \langle x, Ax \rangle - b^T x + c$.

Matrix calculus. Example 2

i Example

Find $df, \nabla f(x)$, if $f(x) = \ln \langle x, Ax \rangle$.

1. It is essential for A to be positive definite, because it is a logarithm argument. So, $A \in \mathbb{S}_{++}^n$. Let's find the differential first:

$$\begin{aligned} df &= d(\ln \langle x, Ax \rangle) = \frac{d(\langle x, Ax \rangle)}{\langle x, Ax \rangle} = \frac{\langle dx, Ax \rangle + \langle x, d(Ax) \rangle}{\langle x, Ax \rangle} = \\ &= \frac{\langle Ax, dx \rangle + \langle x, Adx \rangle}{\langle x, Ax \rangle} = \frac{\langle Ax, dx \rangle + \langle A^T x, dx \rangle}{\langle x, Ax \rangle} = \frac{\langle (A + A^T)x, dx \rangle}{\langle x, Ax \rangle} \end{aligned}$$

Matrix calculus. Example 2

i Example

Find $df, \nabla f(x)$, if $f(x) = \ln \langle x, Ax \rangle$.

1. It is essential for A to be positive definite, because it is a logarithm argument. So, $A \in \mathbb{S}_{++}^n$. Let's find the differential first:

$$\begin{aligned} df &= d(\ln \langle x, Ax \rangle) = \frac{d(\langle x, Ax \rangle)}{\langle x, Ax \rangle} = \frac{\langle dx, Ax \rangle + \langle x, d(Ax) \rangle}{\langle x, Ax \rangle} = \\ &= \frac{\langle Ax, dx \rangle + \langle x, Adx \rangle}{\langle x, Ax \rangle} = \frac{\langle Ax, dx \rangle + \langle A^T x, dx \rangle}{\langle x, Ax \rangle} = \frac{\langle (A + A^T)x, dx \rangle}{\langle x, Ax \rangle} \end{aligned}$$

2. Note, that our main goal is to derive the form $df = \langle \cdot, dx \rangle$

$$df = \left\langle \frac{2Ax}{\langle x, Ax \rangle}, dx \right\rangle$$

Hence, the gradient is $\nabla f(x) = \frac{2Ax}{\langle x, Ax \rangle}$

Matrix calculus. Example 3

Example

Find $df, \nabla f(X)$, if $f(X) = \langle S, X \rangle - \log \det X$.