Pseudo-inverse

By Gemini AI

Here is a detailed explanation of the pseudo-inverse, its computation, and its uses.

What is the Pseudo-inverse?

The **pseudo-inverse**, most commonly the **Moore-Penrose pseudo-inverse**, is a generalization of the standard matrix inverse. While a regular inverse (A^{-1}) only exists for square, non-singular (invertible) matrices, a pseudo-inverse ($A^\dagger$) exists for *any* $m \times n$ matrix, including rectangular or singular (non-invertible) matrices.

If a matrix A *is* invertible, its pseudo-inverse is identical to its regular inverse ($A^\dagger = A^{-1}$).

Pseudo-inverse vs. Regular Inverse

The key differences can be summarized in this table:

Feature	Regular Inverse (A^{-1})	Moore-Penrose Pseudo-inverse ($A^\dagger$)
Existence	Only exists for **square** ($n \times n$) matrices that are **non-singular** (i.e., determinant is non-zero).	Exists for **any** $m \times n$ matrix (square, rectangular, singular, or non-singular).
Property	Satisfies $AA^{-1} = A^{-1}A = I$, where I is the identity matrix.	Satisfies a set of four criteria (the Moore-Penrose conditions). It acts as a "best fit" inverse.
Uniqueness	If it exists, it is unique.	If it exists (which it always does), it is unique.

How to Compute the Pseudo-inverse

There are several ways to compute the pseudo-inverse, depending on the properties of the matrix and the tools you are using.

1. By Matrix Operations (General Case via SVD) The most general and numerically stable method uses **Singular Value Decomposition (SVD)**. Any $m \times n$ matrix A can be decomposed as:

$A = U\Sigma V^T$

Where:

- U is an $m \times m$ orthogonal matrix.
- Σ is an $m \times n$ diagonal matrix of singular values.
- V^T is an $n \times n$ orthogonal matrix (transpose of V).

To find the pseudo-inverse $A^\dagger$, you compute:

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

Here, $\Sigma^\dagger$ is the pseudo-inverse of Σ. It is found by taking the **reciprocal of all *non-zero*** singular values on the diagonal and then transposing the resulting matrix.

2. By Matrix Operations (Special Full-Rank Cases) Simpler formulas exist if the matrix A has **full rank** (all its columns or rows are linearly independent).

- **Full Column Rank (m > n, "tall" matrix):** This is the case for overdetermined systems, like typical least-squares problems. The pseudo-inverse is a **left inverse** ($A^\dagger A = I$). The formula is: $A^\dagger = (A^T A)^{-1} A^T$

- **Full Row Rank (m < n, "wide" matrix):** This is the case for underdetermined systems. The pseudo-inverse is a **right inverse** ($AA^\dagger = I$). The formula is: $A^\dagger = A^T(AA^T)^{-1}$

Connection to your image: The formula in your image, $A^\dagger = R^{-1}Q^T$, is a computationally efficient way to solve the **full column rank** case. It comes from the QR factorization ($A = QR$). If $A = QR$ and A has full column rank, then: $A^\dagger = (A^T A)^{-1} A^T = ((QR)^T(QR))^{-1}(QR)^T = (R^T Q^T QR)^{-1}(R^T Q^T)$ Since Q is orthogonal, $Q^T Q = I$, so this simplifies to: $A^\dagger = (R^T R)^{-1} R^T Q^T = R^{-1}(R^T)^{-1} R^T Q^T = \mathbf{R^{-1}Q^T}$ This is exactly the formula mentioned in your text, which is often used to solve least-squares problems as taught in courses like "VMLS" (likely "Vectors, Matrices, and Least Squares" by Boyd and Vandenberghe).

3. With Julia Functions As your image correctly states, Julia's `LinearAlgebra` standard library provides a direct function:

```
using LinearAlgebra

# Example of a non-square matrix (3x2)
A = [1 2;
     3 4;
```

```
    5 6]

# Compute the pseudo-inverse
A_pinv = pinv(A)
```

The `pinv(A)` function handles all cases (full rank or not) and typically uses the robust SVD method internally.

Use Cases for the Pseudo-inverse

The primary use of the pseudo-inverse is to solve the linear system of equations $Ax = b$ when a unique solution does not exist.

1. **Overdetermined Systems (Least Squares):**
   - **Problem:** You have more equations than unknowns (e.g., fitting a line to 100 data points). There is no exact solution x that satisfies all equations perfectly.
   - **Solution:** $x = A^\dagger b$ provides the **least-squares solution**. This is the vector x that minimizes the error, specifically minimizing the squared Euclidean norm of the residual: $\|Ax - b\|^2$. This is the foundation of linear regression.
2. **Underdetermined Systems (Minimum Norm):**
   - **Problem:** You have fewer equations than unknowns (e.g., $2x + 3y = 5$). There are infinitely many solutions.
   - **Solution:** $x = A^\dagger b$ provides the **minimum-norm solution**. Of all the infinite possible solutions, this is the unique solution x that has the smallest magnitude (smallest Euclidean norm $\|x\|^2$).

It is also used in signal processing, control systems, and image restoration.

The Symbol $A^\dagger$

- **What it is:** The symbol † is called the **"dagger"** (or sometimes "obelisk").
- **Background:** The symbol is used to denote the **Moore-Penrose pseudo-inverse**, named after its originators, **E. H. Moore** (who introduced it around 1920) and **Roger Penrose** (who independently rediscovered and popularized it in 1955). Penrose defined it as the unique matrix that satisfies four specific algebraic properties.
- **Potential Confusion:** In physics and some areas of mathematics, the † symbol is also used to represent the **conjugate transpose** (or Hermitian conjugate) of a complex matrix. This is a *different operation* (though related). However, in the context of linear algebra and solving systems, $A^\dagger$ almost universally refers to the Moore-Penrose pseudo-inverse.

This video discusses the pseudo-inverse in the context of numerical linear algebra and its relation to the SVD. The Pseudoinverse - Numerical Linear Algebra http://googleusercontent.com/youtube_content/0