Partial Least Squares Regression (PLSR): A Detailed Breakdown

Partial Least Squares (PLS) is an extremely powerful technique—similar to PCR—but with a **major improvement**:

> **It uses information from the target variable while constructing components.**

Let's break it down clearly and thoroughly.

What is Partial Least Squares (PLS)?

Partial Least Squares Regression (PLSR) is a supervised dimensionality-reduction + regression method used when:

- You have **many correlated predictors**
- You need to reduce dimensionality
- You want components that are **directly useful for predicting the target**

It works especially well when:

- Predictors are highly collinear
- Number of predictors $\gg$ number of observations
- Standard linear regression becomes unstable

PLS is most common in chemometrics, bioinformatics, spectroscopy, etc.

How is PLS Different from PCA & PCR?

PCA / PCR

- Finds directions of **maximum variance in X only**
- Completely ignores the target ($\mathbf{y}$)
- Dimensionality reduction is unsupervised

PLS

- Finds directions that maximize **covariance between X and y**
- Actively uses the target variable while building components
- Dimensionality reduction is supervised

Key Insight:

PCR components may not be good predictors if they have low variance but high correlation with $\mathbf{y}$. **PLS fixes this by focusing on what actually predicts the response.**

Intuition Behind PLS

Imagine you have 50 features, but only a combination of 3 of them actually matters for predicting your target.

- PCA would find components that maximize variance, not prediction strength.
- PLS finds directions in feature space that **best explain** $\mathbf{y}$.

It's like asking:

> "Which direction in the $\mathbf{X}$ space helps us predict $\mathbf{y}$ the most?"

How PLS Works (Step-by-Step)

Let's define:

- $\mathbf{X}$: predictor matrix $(n \times p)$
- $\mathbf{y}$: response vector $(n \times 1)$
- We want components $\mathbf{t}_1, \mathbf{t}_2, \dots$

PLS builds components iteratively:

Step 1: Compute weights

Find weight vector $\mathbf{w}$ that maximizes:

$$\operatorname{cov}(\mathbf{X}\mathbf{w}, \mathbf{y})$$

This selects the direction in $\mathbf{X}$ that has the strongest predictive relationship with $\mathbf{y}$.

Step 2: Compute the score (component)

$$\mathbf{t} = \mathbf{X}\mathbf{w}$$

This is the new transformed feature.

Step 3: Regress $\mathbf{y}$ on $\mathbf{t}$

$$c = \frac{\mathbf{t}^T\mathbf{y}}{\mathbf{t}^T\mathbf{t}}$$

Step 4: Deflate X and $\mathbf{y}$

Remove the information explained by the component:

$$\mathbf{X} \leftarrow \mathbf{X} - \mathbf{t}\mathbf{p}^T$$

$$\mathbf{y} \leftarrow \mathbf{y} - \mathbf{t}c$$

(where $\mathbf{p}^T$ is the $\mathbf{X}$ loading vector).

Now repeat for the next components.

The result

- You get **latent variables** $\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_k$
- Then you regress $\mathbf{y}$ on these latent variables
- Similar to PCR but the components are **optimal for prediction**

Why PLS Often Outperforms PCR

Because PLS uses **supervised** extraction of components, it tends to:

- ✓ Use fewer components
- ✓ Achieve higher predictive accuracy
- ✓ Avoid throwing away low-variance but high-predictive features
- ✓ Handle multicollinearity better
- ✓ Be more stable when $p \gg n$

PLS vs PCR Summary Table

Table 1: Comparison of PLS and PCR

Feature	**PCA/PCR**	**PLS**
Uses $\mathbf{y}$ during dimensionality reduction?	✗ No	✓ Yes
Maximizes	Variance in $\mathbf{X}$	Covariance($\mathbf{X}$, $\mathbf{y}$)
Risk	Keeps components that may not predict	Components designed to predict
Performance	Good, but sometimes unstable	Typically better, especially with
Latent features	Unsupervised	Supervised

When Should You Use PLS?

Use PLS when:

- You have **many correlated predictors**
- You want to reduce dimensionality **and** predict something
- You need a stable regression model
- You're analyzing chemical spectra, genetics, or high-dimensional datasets

Examples:

- NIR/IR spectroscopy analysis
- Gene expression → disease prediction
- High-dimensional sensor data
- Multicollinearity-heavy datasets

A Simple Analogy

PCR:

> "I'm going to summarize the predictors in the best way possible, even if those summaries don't help predict $\mathbf{y}$ very much."

PLS:

> "I'm going to summarize the predictors in the best way *for predicting* $\mathbf{y}$, even if those summaries don't capture all the variance."
