Linear Algebra for Machine Learning

Mini-Lecture: The Mathematical Foundation

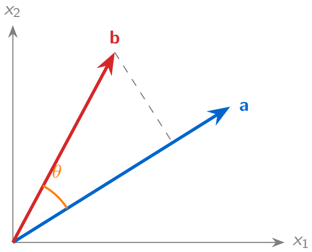
Methods and Algorithms

MSc Data Science

XKCD #1838 by Randall Munroe (CC BY-NC 2.5)

Vectors: Direction and Magnitude

- A **vector** $\mathbf{x} \in \mathbb{R}^n$ represents one data point — e.g. a loan applicant: [income, age, debt_ratio, credit_score]
- **Dot product** $\mathbf{a}^\top \mathbf{b} = \sum_i a_i b_i$ measures similarity between two feature vectors
- **Norm** $\|\mathbf{x}\| = \sqrt{\mathbf{x}^\top \mathbf{x}}$ measures magnitude — large norm = extreme observation



Every ML algorithm starts with data encoded as vectors — master this representation.

Matrices: Data as a Table

- The **design matrix** $\mathbf{X} \in \mathbb{R}^{n \times p}$: n observations (rows), p features (columns)
- Each row is one customer; each column is one attribute (income, age, . . .)
- Matrix–vector product computes **all predictions at once**:

$$\hat{\mathbf{y}} = \mathbf{X}\boldsymbol{\beta} = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{pmatrix} \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}$$

- Finance example: 500 stocks $\times$ 10 factors $\Rightarrow$ **factor exposure matrix**

Think of $\mathbf{X}\boldsymbol{\beta}$ as "apply the model to every observation simultaneously."

Key Matrix Operations

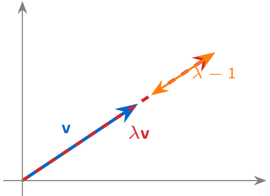
- **Transpose** $\mathbf{A}^\top$: flips rows and columns — needed for $\mathbf{X}^\top\mathbf{X}$
- **Inverse** $\mathbf{A}^{-1}$: "undo" a transformation — $\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$
- The product $\mathbf{X}^\top\mathbf{X}$ is covariance-like and powers the OLS solution (derived in L01)
- $\det(\mathbf{A}) = 0$ means **singular** — no unique inverse, features are linearly dependent

OLS Normal Equation (Preview)

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$$

$\mathbf{X}^\top\mathbf{X}$ is the single most important matrix product in regression — you will see it in L01.

- **Eigen-equation**: $\mathbf{Av} = \lambda \mathbf{v}$
- λ (eigenvalue) = *how much* the vector is stretched
- $\mathbf{v}$ (eigenvector) = *which direction* stays unchanged
- PCA uses eigenvectors of $\mathbf{\Sigma}$ to find principal components (details in L05)



eigenvector stays on its line

Finance Insight

Eigenvalues of a correlation matrix reveal independent **risk factors** — large λ = large variance explained.

Eigenvectors point where data varies most — PCA ranks them by eigenvalue magnitude.

Linear Transformations

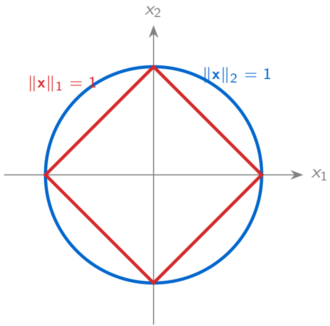
- **Matrix multiplication** = linear transformation of the input space
- $\hat{\mathbf{y}} = \mathbf{X}\beta$ transforms the feature space into a prediction space
- **Rank** of $\mathbf{X}$ = number of linearly independent columns
- If $\text{rank}(\mathbf{X}) < p$: **multicollinearity** — OLS becomes unstable

Practical Check

$$\text{rank}(\mathbf{X}) = p \iff \mathbf{X}^\top\mathbf{X} \text{ is invertible} \iff \text{unique OLS solution exists}$$

Multicollinearity is the #1 numerical pitfall in linear regression — regularization (L01) fixes it.

- **L2 norm** (Euclidean): $\|\mathbf{x}\|_2 = \sqrt{\sum x_i^2}$ — used in Ridge regression
- **L1 norm** (Manhattan): $\|\mathbf{x}\|_1 = \sum |x_i|$ — used in Lasso (sparse solutions)
- Frobenius norm for matrices: $\|\mathbf{A}\|_F = \sqrt{\sum_{ij} a_{ij}^2}$
- Finance: tracking error $= \|\mathbf{r}_{\text{port}} - \mathbf{r}_{\text{bench}}\|_2$



L1 produces sparse models (feature selection); L2 shrinks all coefficients evenly.

- **Covariance matrix**: $\Sigma_{ij} = \mathrm{Cov}(R_i, R_j)$ captures pairwise return co-movements
- **Portfolio variance**: $\sigma_p^2 = \mathbf{w}^\top \boldsymbol{\Sigma}\, \mathbf{w}$, where $\mathbf{w}$ = asset weights
- Eigendecomposition of $\boldsymbol{\Sigma}$ reveals independent **risk factors**

$$\boldsymbol{\Sigma} = \mathbf{V}\,\boldsymbol{\Lambda}\,\mathbf{V}^\top \qquad \text{where} \quad \boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_p)$$

- **Cholesky decomposition** $\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^\top$ generates correlated Monte Carlo samples

Nearly every risk model in quantitative finance starts with $\boldsymbol{\Sigma}$ and its eigenstructure.

Summary: Linear Algebra Toolkit for ML

1. **Vectors** are data points; **matrices** are datasets — $\mathbf{X} \in \mathbb{R}^{n \times p}$
2. $\mathbf{X}^\top \mathbf{X}$ powers the OLS solution and appears throughout regression (L01)
3. **Eigendecomposition** reveals variance directions — foundation of PCA (L05)
4. **Norms** (L1, L2) drive regularization and distance-based methods (L01, L03)

Coming Up

P02: Supervised & Unsupervised Learning — the two paradigms that organize this entire course.

These four ideas — vectors, $\mathbf{X}^\top\mathbf{X}$, eigenvalues, norms — recur in every lecture.