Mathematical Conventions and Fundamentals

Complete Guide to Mathematical Notation and Concepts

ES335 - Machine Learning
IIT Gandhinagar

July 22, 2025

Abstract

This comprehensive tutorial covers essential mathematical notation, conventions, and fundamental concepts needed for machine learning. It includes basic mathematical notation (scalars, vectors, matrices), advanced mathematical operations (derivatives, norms, matrix operations), evaluation metrics for ML, and practical exercises. Understanding these concepts is crucial for following lectures, reading research papers, and implementing algorithms.

Contents

1 Scalar Notation

1.1 Basic Scalars

A **scalar** is a single number. We typically use lowercase letters to denote scalars:

- a, b, c - generic scalars
- α, β, γ - Greek letters for parameters
- n, m, d - dimensions and counts
- ϵ - small positive number (tolerance)
- λ - regularization parameter

Example

If we have a learning rate $\alpha = 0.01$ and a regularization parameter $\lambda = 0.1$, both are scalars.

1.2 Special Scalars

- y_i - the i-th target value (scalar output)
- $\hat{y}_i$ - predicted value for the i-th sample
- $\ell(\hat{y}_i, y_i)$ - loss function (returns a scalar)

2 Vector Notation

2.1 Vector Representation

A **vector** is an ordered list of numbers. We use lowercase bold letters or arrows:

- $\mathbf{x}, \mathbf{y}, \mathbf{z}$ - generic vectors
- $\mathbf{w}$ - weight vector (parameters)
- $\boldsymbol{\theta}$ - parameter vector
- $\boldsymbol{\alpha}$ - vector of dual variables

Vector Examples

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \in \mathbb{R}^d \tag{1}$$

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} \in \mathbb{R}^3 \tag{2}$$

2.2 Vector Operations

- $\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{d} x_i y_i$ - dot product (inner product)
- $\|\mathbf{x}\|$ - norm of vector $\mathbf{x}$
- $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{d} x_i^2}$ - L2 norm (Euclidean norm)
- $\|\mathbf{x}\|_1 = \sum_{i=1}^{d} |x_i|$ - L1 norm (Manhattan norm)
- $\mathbf{x}^T$ - transpose of vector (row vector)

3 Matrix Notation

3.1 Matrix Representation

A **matrix** is a rectangular array of numbers. We use uppercase bold letters:

- $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ - generic matrices
- $\mathbf{A}, \mathbf{B}$ - coefficient matrices
- $\mathbf{I}$ - identity matrix
- $\mathbf{\Sigma}$ - covariance matrix

Matrix Example

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nd} \end{bmatrix} \in \mathbb{R}^{n \times d} \tag{3}$$

where n is the number of samples and d is the number of features.

3.2 Matrix Operations

- $\mathbf{A}^T$ - transpose of matrix $\mathbf{A}$
- $\mathbf{A}^{-1}$ - inverse of matrix $\mathbf{A}$ (if it exists)
- $\mathbf{AB}$ - matrix multiplication
- $\text{tr}(\mathbf{A})$ - trace of matrix $\mathbf{A}$
- $\det(\mathbf{A})$ - determinant of matrix $\mathbf{A}$

4 Common Mathematical Spaces

4.1 Real Number Spaces

- $\mathbb{R}$ - the set of all real numbers
- $\mathbb{R}^d$ - d-dimensional real vector space
- $\mathbb{R}^{n \times d}$ - space of $n \times d$ real matrices

- $\mathbb{R}_+$ - positive real numbers
- $\mathbb{R}_{++}$ - strictly positive real numbers

Space Examples

- A house price: $p \in \mathbb{R}_+$ (positive real number)
- Feature vector: $\mathbf{x} \in \mathbb{R}^d$ (d-dimensional vector)
- Dataset: $\mathbf{X} \in \mathbb{R}^{n \times d}$ (n samples, d features)

4.2 Other Important Spaces

- $\{0, 1\}$ - binary values
- $\{1, 2, \ldots, k\}$ - discrete classes (for k-class classification)
- $[0, 1]$ - unit interval (for probabilities)

5 Dataset and ML-Specific Notation

5.1 Dataset Representation

- $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ - dataset with n samples
- $\mathbf{x}_i \in \mathbb{R}^d$ - i-th feature vector (input)
- $y_i \in \mathbb{R}$ - i-th target value (for regression)
- $y_i \in \{1, 2, \ldots, k\}$ - i-th class label (for classification)
- $\hat{y}_i$ - predicted output for sample i

5.2 Model Parameters

- $\boldsymbol{\theta} \in \mathbb{R}^p$ - parameter vector with p parameters
- $\mathbf{w} \in \mathbb{R}^d$ - weight vector
- $b \in \mathbb{R}$ - bias term (intercept)
- $f(\mathbf{x}; \boldsymbol{\theta})$ - model function parameterized by $\boldsymbol{\theta}$

6 Exercises

6.1 Theoretical Exercises

Exercise #1: Vector Operations

Given vectors $\mathbf{x} = [2, -1, 3]^T$ and $\mathbf{y} = [1, 4, -2]^T$:

(a) Calculate the dot product $\mathbf{x} \cdot \mathbf{y}$

(b) Find the L2 norm $\|\mathbf{x}\|_2$

(c) Compute the L1 norm $\|\mathbf{y}\|_1$

Solutions:

(a) $\mathbf{x} \cdot \mathbf{y} = 2(1) + (-1)(4) + 3(-2) = 2 - 4 - 6 = -8$

(b) $\|\mathbf{x}\|_2 = \sqrt{2^2 + (-1)^2 + 3^2} = \sqrt{4 + 1 + 9} = \sqrt{14}$

(c) $\|\mathbf{y}\|_1 = |1| + |4| + |-2| = 1 + 4 + 2 = 7$

Exercise #2: Matrix Dimensions

For the following expressions, determine if they are valid and find the dimensions:

(a) $\mathbf{A} \in \mathbb{R}^{3\times4}$, $\mathbf{B} \in \mathbb{R}^{4\times2}$. What is the dimension of $\mathbf{AB}$?

(b) $\mathbf{x} \in \mathbb{R}^5$, $\mathbf{W} \in \mathbb{R}^{3\times5}$. What is the dimension of $\mathbf{Wx}$?

(c) $\mathbf{x} \in \mathbb{R}^d$, $\mathbf{y} \in \mathbb{R}^d$. What is the dimension of $\mathbf{x}^T\mathbf{y}$?

Solutions:

(a) Valid. $\mathbf{AB} \in \mathbb{R}^{3\times2}$

(b) Valid. $\mathbf{Wx} \in \mathbb{R}^3$

(c) Valid. $\mathbf{x}^T\mathbf{y} \in \mathbb{R}$ (scalar)

Exercise #3: ML Notation

Given a dataset with 100 samples and 5 features:

(a) Write the dimensions of the feature matrix $\mathbf{X}$

(b) If this is a regression problem, what are the dimensions of the target vector $\mathbf{y}$?

(c) If the weight vector is $\mathbf{w} \in \mathbb{R}^5$ and bias is $b \in \mathbb{R}$, write the prediction for sample i

Solutions:

(a) $\mathbf{X} \in \mathbb{R}^{100\times5}$

(b) $\mathbf{y} \in \mathbb{R}^{100}$

(c) $\hat{y}_i = \mathbf{w}^T\mathbf{x}_i + b$ or $\hat{y}_i = \mathbf{x}_i^T\mathbf{w} + b$

6.2 Coding Exercises

Coding Problem #1: Vector Operations in Python

Implement the following functions in Python using NumPy:

```
import numpy as np

def compute_dot_product(x, y):
    """Compute dot product of two vectors"""
    # TODO: Implement
    pass

def compute_l2_norm(x):
    """Compute L2 norm of a vector"""
    # TODO: Implement
    pass

def compute_l1_norm(x):
    """Compute L1 norm of a vector"""
    # TODO: Implement
    pass

# Test your functions
x = np.array([2, -1, 3])
y = np.array([1, 4, -2])

print(f"Dot product: {compute_dot_product(x, y)}")
print(f"L2 norm of x: {compute_l2_norm(x)}")
print(f"L1 norm of y: {compute_l1_norm(y)}")
```

Solution:

```
def compute_dot_product(x, y):
    return np.dot(x, y)  # or x @ y

def compute_l2_norm(x):
    return np.linalg.norm(x, ord=2)  # or np.sqrt(np.sum(x**2))

def compute_l1_norm(x):
    return np.linalg.norm(x, ord=1)  # or np.sum(np.abs(x))
```

Coding Problem #2: Dataset Creation

Create a synthetic dataset and verify its dimensions:

```
import numpy as np

def create_regression_dataset(n_samples, n_features, noise_std=0.1):
    """Create a synthetic regression dataset"""
    # TODO: Create feature matrix X
    # TODO: Create true weight vector w
    # TODO: Create target vector y = X @ w + noise
    # Return X, y, w
    pass

# Test with 50 samples, 3 features
X, y, w_true = create_regression_dataset(50, 3)

print(f"X shape: {X.shape}")
print(f"y shape: {y.shape}")
print(f"w_true shape: {w_true.shape}")

# Verify dimensions are correct for linear model
print(f"X @ w_true shape: {(X @ w_true).shape}")
```

Solution:

```
def create_regression_dataset(n_samples, n_features, noise_std=0.1):
    np.random.seed(42)  # For reproducibility
    X = np.random.randn(n_samples, n_features)
    w = np.random.randn(n_features)
    noise = np.random.normal(0, noise_std, n_samples)
    y = X @ w + noise
    return X, y, w
```

Coding Problem #3: Linear Model Implementation

Implement a simple linear regression model:

```
class LinearRegression:
    def __init__(self):
        self.w = None
        self.b = None

    def fit(self, X, y):
        """Fit linear regression using normal equation"""
        # Add bias column to X
        # TODO: Implement normal equation solution
        pass

    def predict(self, X):
        """Make predictions"""
        # TODO: Implement prediction
        pass

    def mse(self, X, y):
        """Compute mean squared error"""
        # TODO: Implement MSE calculation
        pass

# Test your implementation
X, y, _ = create_regression_dataset(100, 2)
model = LinearRegression()
model.fit(X, y)
predictions = model.predict(X)
error = model.mse(X, y)
print(f"MSE: {error}")
```

7 Common Mistakes and Tips

7.1 Notation Mistakes to Avoid

1. **Scalar vs Vector confusion:** Always check if you're dealing with scalars (lowercase) or vectors (bold lowercase)
2. **Matrix dimension mismatch:** Always verify matrix multiplication dimensions: $(m \times n) \times (n \times p) = (m \times p)$
3. **Transpose confusion:** Remember $\mathbf{x}^T \in \mathbb{R}^{1 \times d}$ is a row vector, $\mathbf{x} \in \mathbb{R}^d$ is a column vector
4. **Index notation:** x_i is the i-th element, $\mathbf{x}_i$ is the i-th vector

7.2 Best Practices

1. Always write down matrix/vector dimensions when working through problems
2. Use consistent notation throughout your work
3. When coding, use meaningful variable names that reflect the mathematical notation
4. Verify your implementations with simple test cases where you know the answer

8 Reference Quick Sheet

Notation	Type	Example/Description
a, α, λ	Scalar	Single number
$\mathbf{x}, \mathbf{w}, \boldsymbol{\theta}$	Vector	Column vector in $\mathbb{R}^d$
$\mathbf{X}, \mathbf{A}, \boldsymbol{\Sigma}$	Matrix	2D array in $\mathbb{R}^{n \times d}$
$\mathbb{R}$	Space	Set of real numbers
$\mathbb{R}^d$	Space	d-dimensional real vectors
$\mathbb{R}^{n \times d}$	Space	n×d real matrices
$\mathbf{x} \cdot \mathbf{y}$	Operation	Dot product (inner product)
$\|\mathbf{x}\|$	Operation	Vector norm
$\mathbf{A}^T$	Operation	Matrix transpose
$\mathcal{D}$	Set	Dataset
$y_i, \hat{y}_i$	Scalar	True/predicted output

Table 1: Mathematical Notation Reference

9 Advanced Mathematical Operations

9.1 Vector and Matrix Norms

Vector Norms:

- **L1 Norm (Manhattan):** $\|\mathbf{x}\|_1 = \sum_{i=1}^{d} |x_i|$
- **L2 Norm (Euclidean):** $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{d} x_i^2}$
- **L∞ Norm (Maximum):** $\|\mathbf{x}\|_\infty = \max_i |x_i|$
- **General Lp Norm:** $\|\mathbf{x}\|_p = \left(\sum_{i=1}^{d} |x_i|^p\right)^{1/p}$

Matrix Norms:

- **Frobenius Norm:** $\|\mathbf{A}\|_F = \sqrt{\sum_{i,j} a_{ij}^2}$
- **Spectral Norm:** $\|\mathbf{A}\|_2$ = largest singular value

9.2 Matrix Operations and Properties

Basic Operations:

- **Transpose:** $(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$
- **Sum of squares:** $\sum_{i=1}^{n} \epsilon_i^2 = \boldsymbol{\epsilon}^T\boldsymbol{\epsilon}$
- **Scalar property:** For scalar s: $s = s^T$

Matrix Rank and Invertibility:

- **Rank:** Maximum number of linearly independent rows/columns
- **Full Rank:** $\text{rank}(\mathbf{A}) = \min(m, n)$ for $\mathbf{A} \in \mathbb{R}^{m \times n}$
- **Invertible:** Square matrix $\mathbf{A}$ is invertible if $\text{rank}(\mathbf{A}) = n$
- **Singular:** Matrix with no inverse (determinant = 0)

9.3 Calculus for Machine Learning

Derivative of Scalar w.r.t. Vector: If s is a scalar and $\boldsymbol{\theta} \in \mathbb{R}^n$:

$$\frac{\partial s}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial s}{\partial \theta_1} \\ \frac{\partial s}{\partial \theta_2} \\ \vdots \\ \frac{\partial s}{\partial \theta_n} \end{bmatrix}$$

Important Derivative Rules:

- **Linear form:** $\frac{\partial}{\partial \boldsymbol{\theta}}(\mathbf{A}\boldsymbol{\theta}) = \mathbf{A}^T$
- **Quadratic form:** $\frac{\partial}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}^T \mathbf{Z} \boldsymbol{\theta}) = 2\mathbf{Z}\boldsymbol{\theta}$ (when $\mathbf{Z}^T = \mathbf{Z}$)
- **Norm squared:** $\frac{\partial}{\partial \boldsymbol{\theta}} \|\boldsymbol{\theta}\|^2 = 2\boldsymbol{\theta}$

10 Machine Learning Metrics and Evaluation

10.1 Classification Metrics

For classification problems with predictions $\hat{\mathbf{y}}$ and true labels $\mathbf{y}$:

Basic Metrics:

- **Accuracy:** $\frac{|\{i: y_i = \hat{y}_i\}|}{n} = \frac{\sum_{i=1}^{n} \mathbf{1}[y_i = \hat{y}_i]}{n}$
- **Error Rate:** $1 - \text{Accuracy}$

Confusion Matrix Metrics: For binary classification (Positive/Negative classes):

	Predicted +	**Predicted -**
Actual +	TP	FN
Actual -	FP	TN

- **Precision:** $P = \frac{TP}{TP+FP}$ ("Of predicted positives, how many are correct?")
- **Recall (Sensitivity):** $R = \frac{TP}{TP+FN}$ ("Of actual positives, how many found?")
- **Specificity:** $\frac{TN}{TN+FP}$ ("Of actual negatives, how many found?")
- **F1-Score:** $F_1 = \frac{2PR}{P+R} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$

10.2 Regression Metrics

For regression with predictions $\hat{\mathbf{y}}$ and true values $\mathbf{y}$:

- **Mean Squared Error:** $\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$
- **Root Mean Squared Error:** $\text{RMSE} = \sqrt{\text{MSE}}$
- **Mean Absolute Error:** $\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$
- **Mean Error:** $\text{ME} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)$ (can cancel out!)
- **R-squared:** $R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$

11 Advanced Exercises

11.1 Mathematical Operations Practice

Exercise #4: Matrix Derivatives

Given $\boldsymbol{\theta} = [\theta_1, \theta_2]^T$ and $\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}$:

(a) Calculate $\frac{\partial}{\partial \boldsymbol{\theta}}(\mathbf{A}\boldsymbol{\theta})$

(b) Calculate $\frac{\partial}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}^T \mathbf{A}\boldsymbol{\theta})$

(c) Verify $\mathbf{A}^T = \mathbf{A}$ and explain why this matters

Solutions:

(a) $\frac{\partial}{\partial \boldsymbol{\theta}}(\mathbf{A}\boldsymbol{\theta}) = \mathbf{A}^T = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}$

(b) $\frac{\partial}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}^T \mathbf{A}\boldsymbol{\theta}) = 2\mathbf{A}\boldsymbol{\theta} = 2\begin{bmatrix} 2\theta_1 + \theta_2 \\ \theta_1 + 3\theta_2 \end{bmatrix}$

(c) Yes, $\mathbf{A}$ is symmetric, so the quadratic form derivative rule applies directly

Exercise #5: Norm Calculations

For vector $\mathbf{x} = [3, -4, 0, 5]^T$:

(a) Calculate L1, L2, and L∞ norms

(b) Which norm is most sensitive to outliers and why?

(c) Express $\|\mathbf{x}\|_2^2$ in terms of vector operations

Solutions:

(a) $\|\mathbf{x}\|_1 = 12$, $\|\mathbf{x}\|_2 = \sqrt{50} = 5\sqrt{2}$, $\|\mathbf{x}\|_\infty = 5$

(b) L2 norm (squares amplify large values), then L∞, then L1

(c) $\|\mathbf{x}\|_2^2 = \mathbf{x}^T\mathbf{x} = 50$

11.2 ML Metrics Practice

Exercise #6: Classification Metrics

Given confusion matrix for a binary classifier:

	Pred +	**Pred -**
Actual +	85	15
Actual -	10	90

Calculate: (a) Accuracy (b) Precision (c) Recall (d) F1-score
Solutions:

(a) Accuracy $= (85 + 90)/200 = 87.5\%$

(b) Precision $= 85/(85 + 10) = 89.5\%$

(c) Recall $= 85/(85 + 15) = 85\%$

(d) F1-score $= 2 \times 0.895 \times 0.85/(0.895 + 0.85) = 87.2\%$

12 Reference Quick Sheet

Category	**Notation**	**Meaning**
Scalars	a, b, c	Generic scalars
	α, β, λ	Parameters
	n, m, d	Dimensions
	$y_i, \hat{y}_i$	True/predicted values
Vectors	$\mathbf{x}, \mathbf{y}, \mathbf{z}$	Generic vectors
	$\mathbf{w}, \boldsymbol{\theta}$	Parameters
	$\|\mathbf{x}\|_p$	Lp norm
Matrices	$\mathbf{X}, \mathbf{A}, \mathbf{B}$	Generic matrices
	$\mathbf{I}$	Identity matrix
	$\boldsymbol{\Sigma}$	Covariance matrix
	$\text{rank}(\mathbf{A})$	Matrix rank
Spaces	$\mathbb{R}$	Real numbers
	$\mathbb{R}^d$	d-dimensional vectors
	$\mathbb{R}^{n \times d}$	n×d matrices
ML Metrics	TP, FP, TN, FN	Confusion matrix
	Precision, Recall	Classification metrics
	MSE, RMSE	Regression metrics
	F_1	Harmonic mean of P & R

Table 2: Complete Mathematical Notation Reference

13 Further Reading

- **Linear Algebra:** Gilbert Strang, "Introduction to Linear Algebra"
- **ML Math:** Deisenroth, Faisal, Ong, "Mathematics for Machine Learning"
- **Matrix Calculus:** Magnus & Neudecker, "Matrix Differential Calculus with Applications"
- **Online Resources:** Khan Academy Linear Algebra, 3Blue1Brown Essence of Linear Algebra

- **NumPy Documentation:** `https://numpy.org/doc/stable/`
- **Scikit-learn Metrics:** `https://scikit-learn.org/stable/modules/model_evaluation.html`