Recap: Distances, Norms and Similarity

DS701 Session 3 — Mon Sep 14, 2026

Today’s plan

Knowledge-Check Review

KC 1: What makes a metric?

*State the four properties a function $d(x, y)$ must satisfy to be called a **metric**. Then say, in one line, how a **norm** $p(\cdot)$ gives you a metric for free.*

KC 2: Inner products, norms, angles

Let $\mathbf{u} = [1, 2, 0]$ *and* $\mathbf{v} = [3, -1, 4]$. *Compute (a) the inner product* $\mathbf{u} \cdot \mathbf{v}$, *(b)* $\|\mathbf{u}\|_2$ *and* $\|\mathbf{v}\|_2$, *(c) the cosine of the angle between them, and (d) the* ℓ_1 *and* ℓ_∞ *distances* $\|\mathbf{u} - \mathbf{v}\|_1$, $\|\mathbf{u} - \mathbf{v}\|_\infty$. *Then explain in one sentence why the cosine can be near zero while the vectors are far apart in every* ℓ_p *distance.*

KC 3: Hamming vs. Jaccard

Two situations from the lecture. Case 1: two long documents, almost identical, that differ in 10 words. Case 2: two 5-word documents with no words in common. Represent each document as a bit vector over the vocabulary. What is the Hamming distance in each case, why does that ranking feel wrong, and which measure from the lecture fixes it (give its formula)?

Highlights

Linear-algebra essentials

Everything today lives in $\mathbb{R}^d$: a data object is a vector $\mathbf{x} = [x_1, \dots, x_d]$, a dataset is an $m \times d$ matrix (rows = objects, columns = features).

Inner product — elementwise multiply, then sum:

$$\mathbf{u} \cdot \mathbf{v} = \mathbf{u}^T \mathbf{v} = \sum_i u_i v_i$$

Angle — inner product with the lengths divided out:

$$\cos\theta = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|_2 \, \|\mathbf{v}\|_2}$$

Norm — a length. The ℓ_2 norm is the one you know:

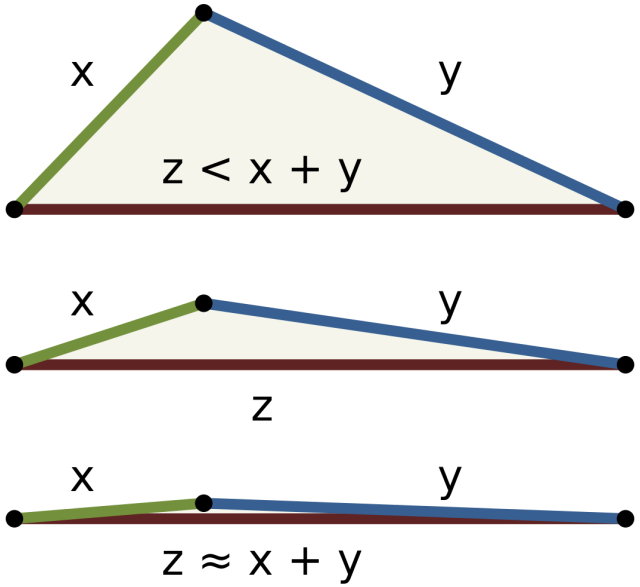
$$\|\mathbf{v}\|_2 = \sqrt{\mathbf{v} \cdot \mathbf{v}} = \sqrt{\textstyle\sum_i v_i^2}$$

Distance — the norm of the difference:

$$d(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\|$$

What makes a metric

A **metric** $d(x, y)$ satisfies



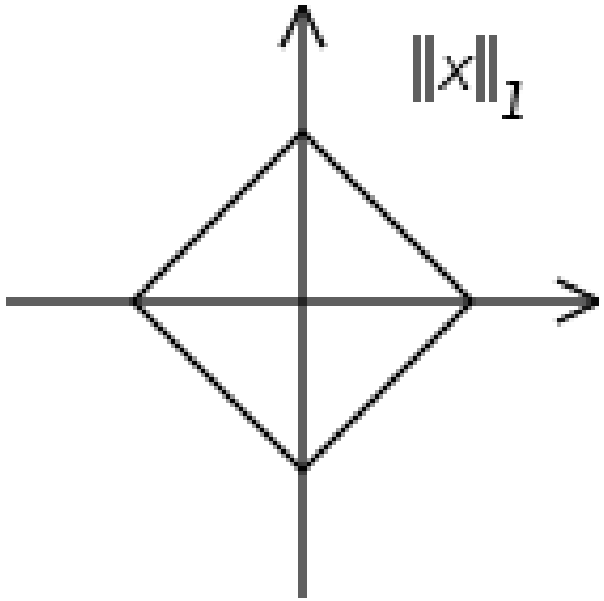
Norm $\longrightarrow$ metric: the ℓ_p family

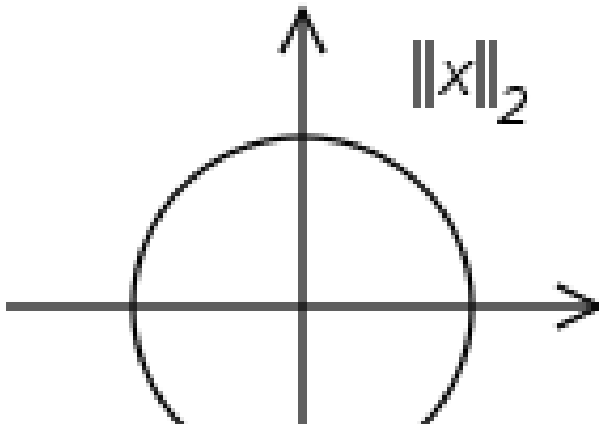
A **norm** $p(\mathbf{v})$: $p(a\mathbf{v}) = |a|\,p(\mathbf{v})$, $p(\mathbf{u}+\mathbf{v}) \le p(\mathbf{u}) + p(\mathbf{v})$, $p(\mathbf{v}) = 0 \iff \mathbf{v} = 0$.

Every norm gives a metric: $d(\mathbf{x}, \mathbf{y}) = p(\mathbf{x} - \mathbf{y})$.

$$\|\mathbf{x} - \mathbf{y}\|_p = \Big(\sum_{i=1}^{d} |x_i - y_i|^p \Big)^{1/p}, \qquad p \ge 1$$

p	norm	distance
1	$\sum_i \lvert x_i - y_i \rvert$	Manhattan (Hamming on bits)
2	$\sqrt{\sum_i (x_i - y_i)^2}$	Euclidean
∞	$\max_i \lvert x_i - y_i \rvert$	Chebyshev /





Which ℓ_p is right?

The choice of p decides **what gets emphasized**.

ℓ_1 — Manhattan

- many small differences count
- robust to a single large one
- natural for sparse data, counts, bit vectors

ℓ_2 — Euclidean

- the default in geometric space
- smooth, differentiable, rotation-invariant
- squares → outliers hurt more than in ℓ_1

ℓ_∞ — Chebyshev

- *only* the largest difference counts
- extremely outlier-sensitive
- right when you need a uniform bound

Cosine vs. Euclidean

Cosine similarity — the angle, lengths divided out:

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^T \mathbf{y}}{\|\mathbf{x}\| \, \|\mathbf{y}\|}$$

- 1 same direction, 0 orthogonal
- **scale-invariant**: $\cos(\mathbf{x}, 2\mathbf{y}) = \cos(\mathbf{x}, \mathbf{y})$
- dissimilarity: $1 - \cos(\mathbf{x}, \mathbf{y})$ — *not* a metric

Euclidean distance — the length of the gap:

$$\|\mathbf{x} - \mathbf{y}\|_2$$

- 0 only when identical
- **scale-sensitive**: doubling $\mathbf{y}$ moves it
- a metric

Why scaling matters

The **units** of each feature are part of the model. Wine dataset, 13 chemical measurements:

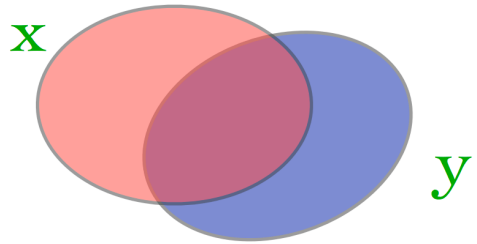
feature	typical range	std. dev.
`hue`	0.5 – 1.7	0.23
`alcohol`	11 – 15	0.81
`magnesium`	70 – 160	14
`proline`	280 – 1680	**314**

Sets and bit vectors: Hamming vs. Jaccard

Bit vector as a **code** (every position matters):
Hamming $= \ell_1 =$ number of bit flips.

Bit vector as a **set** (which items are present):
Hamming counts the set difference, and that misleads — 10 differing words out of thousands beats 5 disjoint words.

Jaccard normalizes by how much there was to agree on:



$$J_{sim} = \frac{|x \cap y|}{|x \cup y|}, \qquad J_{dist} = 1 - J_{sim} \quad (\text{a metric})$$

In-Class Activity

Activity: Distance matrices, by hand and with `scikit-learn`

Goal: build Euclidean, Manhattan, cosine and Jaccard distance matrices on real data with `numpy`, verify them against `sklearn` / `scipy`, watch nearest neighbors change with the metric and with feature scaling — then two stretch lessons: the `fit` / `transform` pattern that every `sklearn` object shares, and a 10-minute **dynamic time warping** teaser (why plain distances fail on time-shifted series).

- Work in groups of 2–3.
- Parts marked (**autograded**) are submitted to Gradescope; the rest is participation.
- Staff will circulate — be ready to explain any part of your work.
- Dependencies: `numpy`, `pandas`, `matplotlib`, `scipy`, `scikit-learn` only.

Section A1: Open in Colab Open on GitHub

Section B1: Open in Colab Open on GitHub

Going deeper

Full lecture notes: Distances and Time Series

- Feature space representation — one-hot encoding and why not 1, 2, 3
- Metrics — the four axioms and the distance matrix
- Norms and the ℓ_p family — Minkowski distance, ℓ_1, ℓ_2, ℓ_∞, and the not-a-norm ℓ_0
- Similarity and dissimilarity — cosine and how to turn a similarity into a dissimilarity
- Bit vectors and sets — Hamming and Jaccard
- Linear-algebra background: dot product and ℓ_2 norm, definitions: unit vector, distance, angle
- The `sklearn` API pattern used in the stretch section: Scikit-Learn appendix
- Time series and dynamic time warping in full: optional module M2 Time Series (and the DTW section of this lecture)