t-Stochastic Neighbor Embedding

Complete 80-Slide Presentation

Prof.Asc. Endri Raco

Polytechnic University of Tirane

October 2025

What is Dimensionality Reduction?

Definition

Transforming high-dimensional data into lower-dimensional representations while preserving meaningful structure

Why We Need It:

- Visualization: Human perception limited to 3D
- Curse of dimensionality: Distance becomes meaningless in high-D
- Computational efficiency: Reduce processing requirements
- Feature extraction: Identify essential patterns

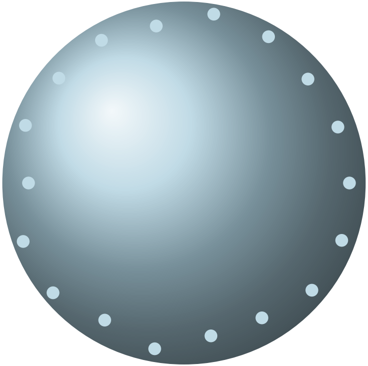
The Central Challenge:

How do we decide what to preserve when we must lose information?

Traditional answer: Preserve distances

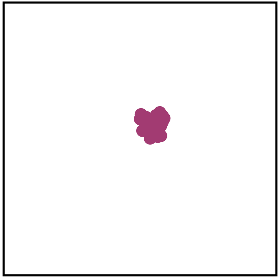
t-SNE answer: Preserve neighborhoods

The Fundamental Challenge of Dimensionality Reduction



784 Dimensions

MNIST digit



2 Dimensions

Your screen

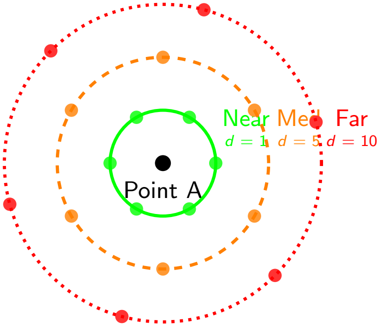
How do we preserve neighborhood relationships

The Crowding Problem: Why Linear Methods Fail

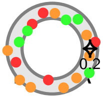
Definition

Crowding Problem: The geometric impossibility of preserving moderate-range distances when projecting from high to low dimensions, causing distinct distance scales to collapse.

High-D Space (10D)



After Linear Projection to 2D



Ratio: 1 : 1.1 : 1.2

The Paradigm Shift: From Geometry to Information

Traditional Methods

Preserve distances or variance

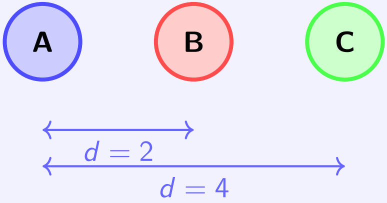
$\downarrow$

t-SNE

Preserve probability distributions

The Paradigm Shift: Concrete Example

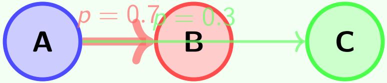
Traditional: Preserve Distances



Problem: All distances treated equally

No context about local density

t-SNE: Preserve Probabilities



Solution: Likelihood encodes context

Adapts to local density automatically

Key Insight: Same distance $\rightarrow$ different probabilities based on neighborhood density

Building Intuition: From Distances to Neighborhoods

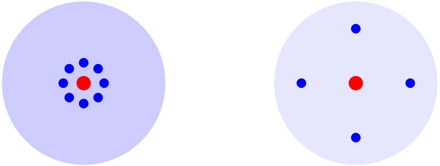
The Problem with Raw Distances:

- Point A: 1 unit from B, 10 units from C
- But what if A is in dense region?
- And C is in sparse region?
- Raw distance loses context!

The Solution - Relative Similarity:

- Convert distances to probabilities
- "How likely is B to be A's neighbor?"
- Adapt to local density automatically
- Use Gaussian decay (smooth, differentiable)

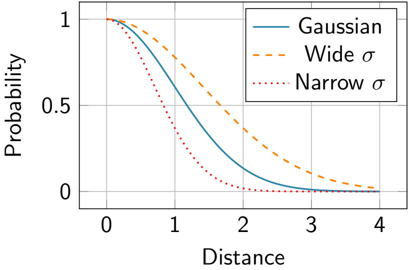
Dense Region **Sparse Region**



Small σ needed Large σ needed

Key Idea: Each point gets its own "neighborhood size" (σ_i) based on local density

From Distances to Probabilities



Key Transformation:

$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-\|x_i - x_k\|^2/2\sigma_i^2)}$$

Insight: σ_i adapts to local density automatic

Why Gaussian? The Natural Choice for Neighborhoods

What We're Building

Our Goal:

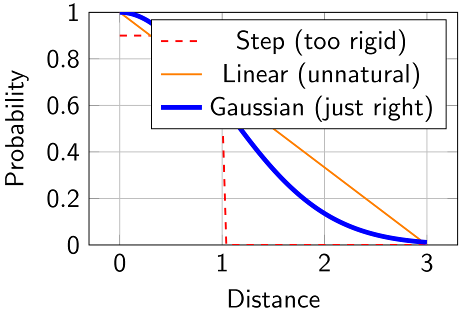
- Convert distances to probabilities
- "How likely is j to be i's neighbor?"
- Must adapt to local density

Three Key Requirements:

1. Smooth decay - no sudden cutoffs
2. Local focus - neighbors matter most
3. Unbiased - don't assume patterns

The Winner: Gaussian $p_{j|i} \propto e^{-d_{ij}^2/2\sigma_i^2}$

Visual Comparison



Analogy: Friendship strength strongest nearby, fading smoothly

The Mathematics Behind Gaussian: Maximum Entropy Principle

The Core Principle

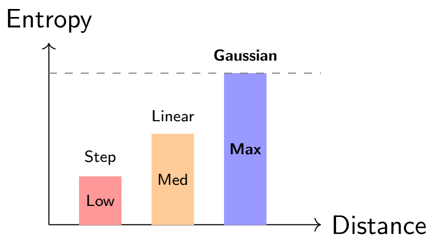
The Question:
Which distribution makes the *fewest assumptions* while matching our constraints?

Answer: Maximum Entropy
The distribution with highest uncertainty (entropy) given the constraints

Why This Matters:

- Most "honest" - no hidden bias
- Adds no assumptions
- Principled approach

Entropy Comparison



Gaussian = Maximum Entropy
Most uncertain = Least biased

The Mathematical Derivation: Problem Setup

Optimization Problem

Maximize Entropy:

$$H(P_i) = -\sum_j p_{j|i} \log p_{j|i}$$

Subject to Constraints:

1. Normalization: $\sum_j p_{j|i} = 1$
2. Fixed Variance: $\sum_j p_{j|i} d_{ij}^2 = \sigma_i^2$

The goal is to find the most unbiased probability distribution ($p_{j|i}$) that meets our constraints.

The Mathematical Derivation: Solution

Solution via Lagrange Multipliers

1. **The Lagrangian:**

$$\mathcal{L} = H(P_i) + \lambda \left(\sum p_{j|i} - 1 \right) + \mu \left(\sum p_{j|i} d_{ij}^2 - \sigma_i^2 \right)$$

2. **Taking derivatives and solving for** $\frac{\partial \mathcal{L}}{\partial p_{j|i}} = 0$ **yields the result.**
3. **Result (The Gaussian Distribution):**

$$p_{j|i} = \frac{e^{-\frac{d_{ij}^2}{2\sigma_i^2}}}{\sum_k e^{-\frac{d_{ik}^2}{2\sigma_i^2}}}$$

Perplexity: Setting the Neighborhood Size

The Problem We're Solving

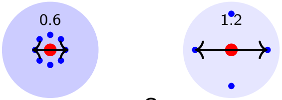
Question: How many neighbors should each point consider?

Challenge: Different regions have different densities!

- Dense areas: Small σ needed
- Sparse areas: Large σ needed

Solution: Perplexity - a user parameter that sets "effective" number of neighbors

Adaptive Neighborhoods



Dense: $\sigma = 0.1$ Sparse: $\sigma = 0.5$

Both have same perplexity $= 5$
Different σ values!

Key Insight: Perplexity is constant across all points, but σ_i adapts to achieve it

Perplexity: The Mathematics and Algorithm

Mathematical Definition

From Shannon Entropy:

$$H(P_i) = -\sum_j p_{j|i} \log_2 p_{j|i}$$

Perplexity:

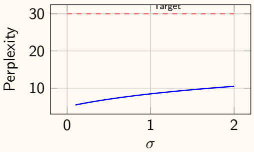
$$\text{Perp}(P_i) = 2^{H(P_i)}$$

Interpretation:

- $\text{Perp} = 5 \rightarrow$ "acts like" 5 neighbors
- $\text{Perp} = 30 \rightarrow$ "acts like" 30 neighbors

Finding σ_i: Binary Search

Why Binary Search?
Perplexity increases with σ monotonically



Algorithm:

1. Start with $\sigma = 1$
2. Compute current perplexity
3. Too high? $\rightarrow$ Decrease σ

Measuring Information Loss: KL Divergence

What is KL Divergence?

$$\mathrm{KL}(P||Q) = \sum_j p_j \log \frac{p_j}{q_j}$$

Extra bits needed when using Q instead of true P

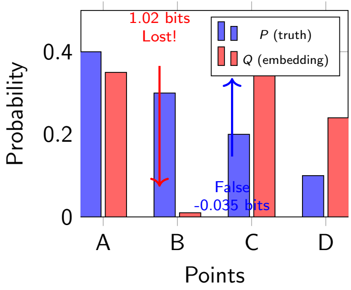
Critical Asymmetry Example

Consider point B with true probability 0.3:

Missing a true neighbor:
True: $p = 0.3$, Embedded: $q = 0.01$
Penalty: $0.3 \times \log(30) \approx$ **1.02 bits**

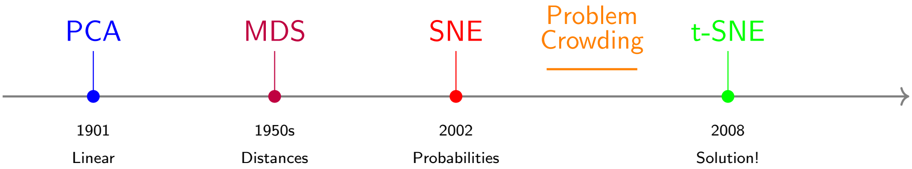
Creating a false neighbor:
True: $p = 0.01$, Embedded: $q = 0.3$
Penalty: $0.01 \times \log(0.033) \approx$ **-0.035 bits**

Visual Example



Original SNE: The Precursor to t-SNE

A Brief History of Dimension Reduction



SNE's Innovation

- First to use probabilities
- Adaptive neighborhoods (σ_i)
- Information-theoretic approach
- KL divergence for optimization

SNE's Fatal Flaw

- Used Gaussian in low-D space
- Cannot represent moderate distances
- Led to "crowding problem"
- All points collapse together

SNE's Mathematics: Where It Went Wrong

The Formulation

- **High-D Similarity (P):**
  Gaussian with adaptive variance σ_i

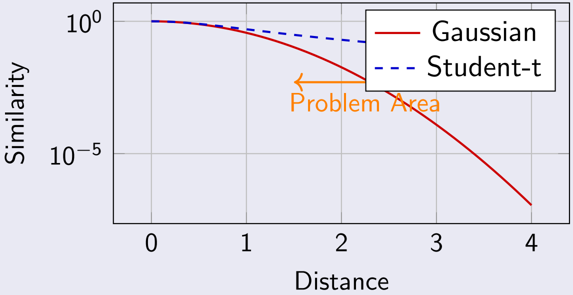
$$p_{j|i} = \frac{\exp(-d_{ij}^2/2\sigma_i^2)}{\sum_k \exp(-d_{ik}^2/2\sigma_i^2)}$$

- **Low-D Similarity (Q):**
  Gaussian with fixed variance

$$q_{j|i} = \frac{\exp(-d_{ij}^2)}{\sum_k \exp(-d_{ik}^2)}$$

- **Cost Function:** $C = \sum_i \mathrm{KL}(P_i || Q_i)$

Why Gaussian Fails in 2D



Problem: Moderate distances in high-D get exponentially tiny similarities in low-D, causing crowding.

The Curse: Why High-D Breaks Our Intuition

The Volume Problem

Question: In a D-dimensional sphere, what fraction of volume is in the outer shell (radius 0.9 to 1.0)?

Your intuition (2D):

$$\frac{\pi \cdot 1^2 - \pi \cdot 0.9^2}{\pi \cdot 1^2} = 19\%$$

Reality in high-D:

- 5D: 41%
- 10D: 65%
- 50D: 99.5%
- **100D: 99.997%**

Volume Distribution by Dimension