t-SNE: Visualizing High-Dimensional Data

Mini-Lecture: Seeing Hidden Structure in Your Data

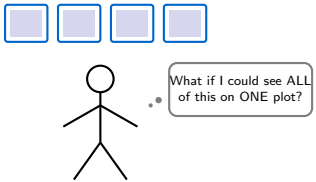
Methods & Algorithms

MSc Data Science – Spring 2026

Why Can't We See 50 Variables at Once?

The Visualization Wall

- You track 1000 stocks with 50 features each (returns, volatility, sector, momentum...)
- A scatter plot needs 2 axes – where do the other 48 go?
- Pairwise plots? That is $\binom{50}{2} = 1225$ scatter plots
- We need a way to compress 50D into 2D *without losing structure*



Human vision is limited to 3D – dimensionality reduction bridges the gap between data complexity and perception

You Have 1000 Stocks and 50 Features

Think About It

- Imagine your data matrix: 1000 rows (stocks) $\times$ 50 columns (features)
- Each stock lives in a 50-dimensional space – impossible to visualize directly
- **Key intuition**: stocks in the same sector probably cluster together in this 50D space
- The clusters are *there* – we just cannot see them

The Question

Is there an algorithm that can project 50D onto 2D while keeping similar stocks close together?

Spoiler: PCA tries to preserve *global variance*. t-SNE tries to preserve *local neighborhoods*. For discovering clusters, neighborhoods matter more.

Dimensionality reduction is not about throwing away data – it is about revealing hidden structure

What IS t-SNE?

t-SNE in Plain English
An algorithm that squeezes high-dimensional data into 2D while preserving which points are **neighbors**.

Key Terms

- **Embedding**: the 2D output – each point gets an (x, y) coordinate
- **Perplexity**: how many neighbors each point "pays attention to" (typically 5–50)
- **Neighborhood**: points that are close in the original 50D space

t-SNE stands for:

t-distributed
Stochastic
Neighbor
Embedding

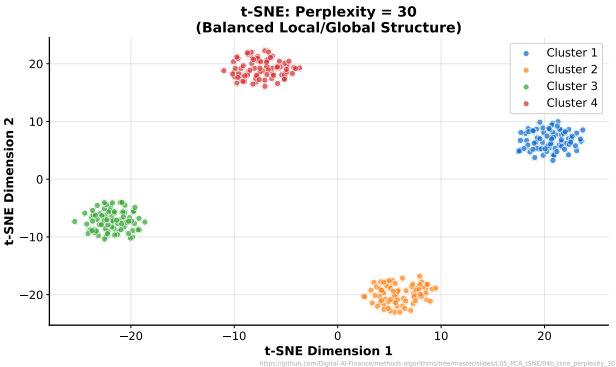
van der Maaten & Hinton (2008)

Core Idea

If two stocks are similar in 50D, they should land near each other in 2D. If they are different, they should be far apart.

t-SNE is for visualization only – do not use the 2D coordinates as features for a downstream model

How Does t-SNE Reveal Hidden Structure?



What You See

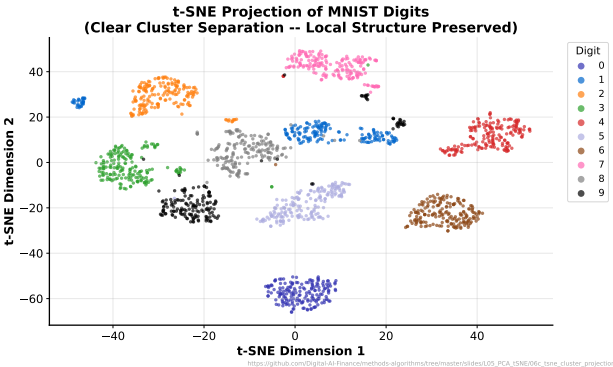
- Each dot is a data point (e.g. a stock described by 50 features)
- Colors represent true groups (sectors, regimes)
- t-SNE places similar points close together in 2D
- Clusters emerge that were invisible in raw data tables

Finance Interpretation

Tech stocks cluster together, banks cluster together, energy stocks cluster together – because their 50-feature profiles are similar.

Perplexity=30 (sklearn default) balances local and global structure – always a good starting point

How Does t-SNE Preserve Neighborhoods?



The Neighborhood Rule

- Points close in high-D stay close in 2D
- Points far apart in high-D get pushed further apart
- The algorithm minimizes the mismatch between high-D and low-D neighborhoods
- Result: clusters that overlap in PCA become separated in t-SNE

Key Difference from PCA

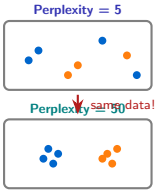
PCA preserves *directions of maximum spread*. t-SNE preserves *who is near whom*.

t-SNE uses a Student-t distribution in low-D to prevent the "crowding problem" – moderate distances get room to spread

What Can Go Wrong with t-SNE?

Parameter Sensitivity

- Different perplexity values produce different cluster layouts
- Different random seeds produce different orientations
- Cluster *sizes* in 2D are meaningless
- Distances *between* clusters are meaningless



Golden Rule

Always run t-SNE with at least 3 different perplexity values. If a pattern disappears when you change perplexity, it is an artifact.

Wattenberg et al. (2016) "How to Use t-SNE Effectively" – required reading before interpreting t-SNE plots

Where Do Quants Use t-SNE?

Four Key Applications in Finance

Application	How It Works
Market Regime Detection	Project daily return vectors of 50 stocks into 2D. Color by date. Clusters = regimes (bull, bear, crisis).
Portfolio Visualization	Map 500 stocks by their 50 risk factors. See which stocks group together without sector labels.
Anomaly Detection	Points far from any cluster in the t-SNE map are potential outliers – flash crashes, rogue trades.
Customer Segmentation	Visualize banking customers by 30+ behavioral features. Discover natural groups for product targeting.

t-SNE is an exploration tool – it generates hypotheses about structure, which you then test with statistical methods

When Should You Use t-SNE vs PCA?

Use Case	PCA	t-SNE
Need new coordinate system	✓	
Preprocessing for ML model	✓	
Interpretable axes (loadings)	✓	
Discovering hidden clusters		✓
Exploring non-linear structure		✓
Presenting to non-technical audience		✓
Speed matters ($n > 10{,}000$)	✓	
Projecting new data points	✓	

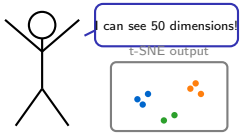
Best Practice

Run PCA first to reduce to 30–50 dimensions, then apply t-SNE on the PCA output. This is faster, more stable, and removes noise.

PCA is a tool (transforms data for modeling). t-SNE is a lens (shows you what the data looks like). Use both.

Summary: t-SNE in 4 Takeaways

1. **t-SNE preserves neighborhoods**: similar points in high-D land near each other in 2D
2. **Perplexity controls the view**: low = local detail, high = global structure. Always try 3+ values.
3. **Do not over-interpret**: cluster sizes, inter-cluster distances, and axis orientation are meaningless
4. **Finance use**: regime detection, anomaly spotting, and portfolio visualization



Next Steps

Explore the deep dive for KL divergence derivation, the crowding problem, and UMAP comparison.

t-SNE is for exploration, not measurement. Use it to generate hypotheses, then test them with rigorous statistics.