L05: PCA & t-SNE

Dimensionality Reduction for Visualization and Preprocessing

Methods and Algorithms

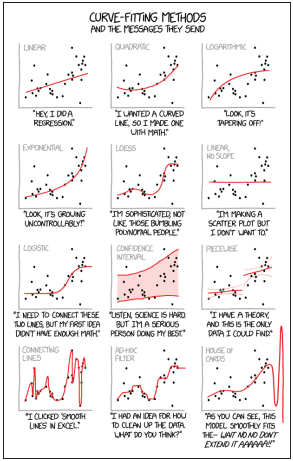
MSc Data Science

Spring 2026

Outline

Three zones: Introduction, Core Content (PMSP), and Wrap-Up

Fitting Curves in High Dimensions



"With enough dimensions, every dataset looks like a straight line."

XKCD #2048 "Curve Fitting" by Randall Munroe (CC BY-NC 2.5)

The Dimensionality Problem

Why high dimensions are a headache

- **Too many features:** a portfolio with 100 assets lives in a 100-dimensional space
- **Beyond human perception:** we cannot visualize anything beyond 3 dimensions
- **Redundant information:** many features are correlated and carry overlapping signals
- **The goal:** compress data to fewer dimensions while preserving essential structure

---

High dimensions cause sparsity, increase computation, and invite overfitting

Two Approaches to Dimensionality Reduction

PCA — Find directions of greatest spread

- Like choosing the best camera angle to photograph a 3D object
- Linear, fast, and mathematically elegant

t-SNE — Preserve which points are neighbors

- Like seating friends together at a wedding reception
- Non-linear, captures complex structure, but visualization only

PCA preserves global variance; t-SNE preserves local neighborhood relationships

Why Banks Care About Dimensionality Reduction

Real applications in finance and banking

- **Yield curves:** 20+ maturities collapse to just 3 interpretable factors
- **Portfolio risk:** hundreds of correlated assets reduce to a few risk drivers
- **Customer analytics:** visual clusters reveal segments in high-dimensional data
- **Market regimes:** embeddings separate calm, volatile, and transition periods

Dimensionality reduction is a daily tool in quantitative finance and risk management

Learning Objectives

By the end of this lecture, you will be able to:

1. **Derive** PCA from variance maximization and explain the SVD–PCA equivalence
2. **Evaluate** dimensionality reduction methods (PCA vs. t-SNE vs. UMAP) for a given dataset
3. **Analyze** the effect of hyperparameters (perplexity, learning rate) on t-SNE embeddings
4. **Critique** PCA assumptions and limitations for nonlinear financial data (e.g., yield curves)

Finance Application: Portfolio risk decomposition, yield curve analysis, market regime detection

---

Bloom's Level 4–5: Analyze, Evaluate, Create

The Business Problem

Curse of Dimensionality in Practice

- **Portfolio management:** 100+ correlated assets make risk estimation unstable
- **Customer profiling:** dozens of behavioral features with heavy redundancy
- **Consequences:** data becomes sparse, models overfit, computation explodes

What we need:

- Compress many features into a few meaningful dimensions
- Preserve the relationships that matter for downstream tasks

---

Dimensionality reduction tackles sparsity, overfitting, and computational cost simultaneously

Key Equations: PCA

Covariance Matrix (from mean-centered data $X_c = X - \bar{X}$):

$$C = \frac{1}{n-1} X_c^\top X_c$$

Eigendecomposition: find directions and magnitudes

$$C\,\mathbf{v}_k = \lambda_k\,\mathbf{v}_k$$

Explained Variance Ratio: how much each component captures

$$\text{EVR}_k = \frac{\lambda_k}{\sum_{j=1}^{p} \lambda_j}$$

PCA reduces to an eigenvalue problem on the covariance matrix

Key Equations: t-SNE

High-dimensional similarity (Gaussian kernel):

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

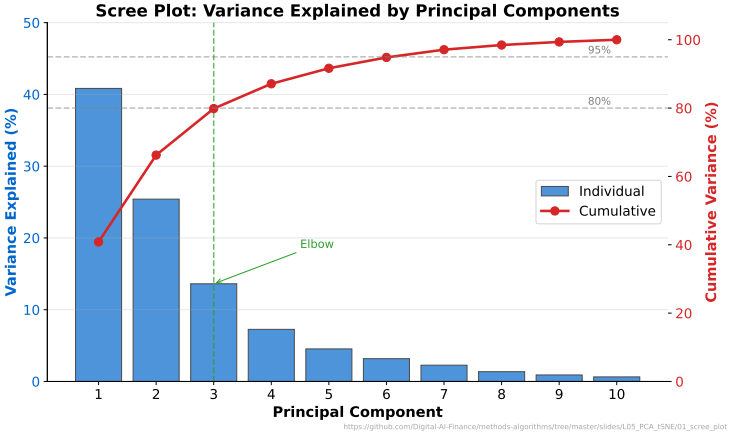
Low-dimensional similarity (Student-t with 1 degree of freedom):

$$q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y_k - y_l\|^2\right)^{-1}}$$

Objective: minimize the Kullback–Leibler divergence $\mathrm{KL}(P \| Q)$

The heavy-tailed t-distribution in low-D prevents crowding of moderate neighbors

Scree Plot: Choosing the Number of Components



Choose k capturing 80–95% variance or at the elbow — Kaiser criterion ($\lambda > 1$) valid for correlation matrix only

Yield Curve PCA: Three Factors Explain 98%

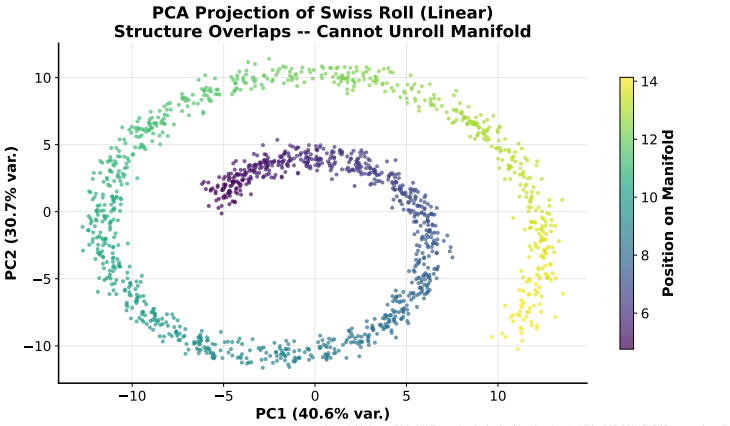
Three principal components capture nearly all yield curve variation:

- **PC1 — Level (∼85%):** all maturities load in the same direction; a parallel shift up or down
- **PC2 — Slope (∼10%):** short and long maturities load with opposite signs; steepening or flattening
- **PC3 — Curvature (∼3%):** short and long load positively, middle negatively; a butterfly twist

Together these three factors explain over 98% of daily yield curve movements.

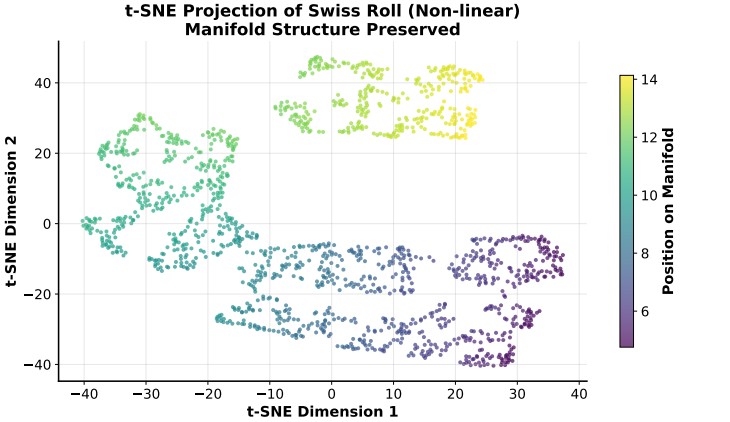
Level–Slope–Curvature decomposition is the foundation of fixed-income risk management

PCA on Swiss Roll: Linear Limits



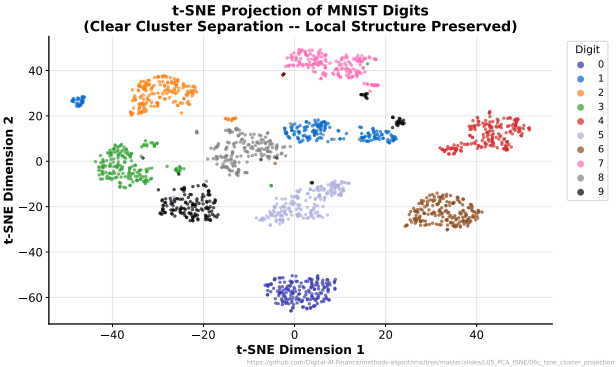
PCA projects linearly and cannot unroll a curved manifold — distant points on the surface overlap

t-SNE on Swiss Roll: Unrolling the Manifold



t-SNE preserves local neighborhoods and successfully unrolls the non-linear manifold structure

t-SNE Cluster Preservation: MNIST Digits



t-SNE on 784-dimensional MNIST data: digit classes form clearly separated clusters in 2D

t-SNE Caveats: What You Cannot Conclude

Common misinterpretations to avoid:

- **Cluster sizes are not meaningful:** visual size depends on local density, not group count
- **Inter-cluster distances are not meaningful:** only within-cluster proximity is preserved
- **Results are non-deterministic:** different random seeds produce different layouts

Best practice pipeline:

- Standardize features, then PCA to 30–50 dimensions, then t-SNE to 2D
- Use t-SNE for **visualization only** — never as input features for downstream models

t-SNE is an exploratory tool, not a preprocessing step — always verify patterns with other methods

PCA vs. t-SNE: Head-to-Head Comparison

Property	PCA	t-SNE
Type	Linear	Non-linear
Speed	Fast ($O(np^2)$)	Slow ($O(n^2)$ naive)
Deterministic	Yes	No (random init)
Preserves	Global variance	Local neighborhoods
Reversible	Yes (approx.)	No
Use for ML features	Yes	No
Visualization	Limited	Excellent

PCA for preprocessing and speed; t-SNE for visualization and exploration

Portfolio Risk Decomposition with PCA

Reducing 100 assets to a few interpretable risk factors:

- **PC1 — Market factor:** broad market movement affecting all assets (largest eigenvalue)
- **PC2–3 — Sector factors:** industry-specific rotations (e.g., tech vs. energy)
- **Higher PCs — Idiosyncratic risk:** asset-specific noise, often discarded

Practical impact:

- Covariance estimation stabilized: estimate 3–5 factor loadings instead of 5,050 correlations
- Regulatory capital models (Basel) use PCA-based factor decompositions

---

PCA-based factor models are the backbone of portfolio risk management and stress testing

Market Regime Detection with t-SNE

Visualizing hidden structure in market data:

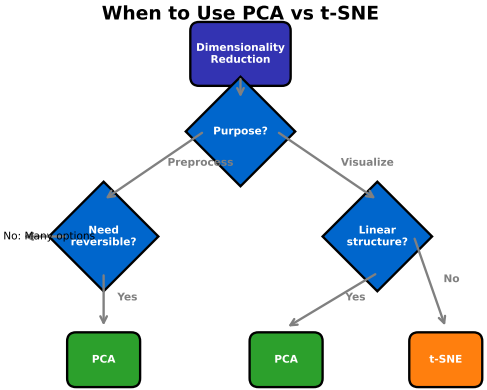
- **Input features:** rolling volatility, correlations, spreads, volumes across asset classes
- **t-SNE projection:** reveals distinct clusters corresponding to market regimes
- **Typical regimes:** calm/low-vol, crisis/high-vol, and transition periods

Application: use detected regimes to build regime-conditional strategies and risk models.

---

t-SNE visualization guides hypothesis generation — confirm regimes with formal clustering

Decision Framework: Which Method to Use?



PCA: Fast, linear, reversible, for preprocessing

t-SNE: Slow, non-linear, visualization only, preserves local structure

https://github.com/Digital-AI-Finance/methods-algorithms/tree/master/slides/L05_PCA_tSNE/07_decision_flowchart

Start with PCA for preprocessing; add t-SNE when you need 2D visualization of complex structure

Hands-on Exercise

Open the Colab Notebook and complete these exercises:

1. **PCA on finance data:** apply PCA to a multi-asset return dataset, plot the scree curve, and interpret the first three components
2. **t-SNE visualization:** embed high-dimensional customer or market data into 2D and identify visual clusters
3. **Method comparison:** run both PCA and t-SNE on the same dataset and discuss what each method reveals and misses

---

Notebooks available on the course GitHub page — see the L05 folder

Key Takeaways

What to remember from this lecture:

- **PCA:** linear, fast, reversible — use for preprocessing, denoising, and factor extraction
- **t-SNE:** non-linear, stochastic — use for 2D/3D visualization only, never as ML features
- **Standard pipeline:** Standardize → PCA (30–50 dims) → t-SNE (2D) for best results
- **Finance:** yield curve decomposition, portfolio factor models, market regime detection

---

Dimensionality reduction is both a preprocessing tool (PCA) and an exploration tool (t-SNE)

Closing Thought

"We reduced our 100-dimensional portfolio to 3 principal components. The fourth component? That's just noise... probably."

— Overheard in a quantitative risk team

References

- Jolliffe, I.T. (2002). *Principal Component Analysis*, 2nd ed. Springer.
- van der Maaten, L. & Hinton, G. (2008). Visualizing Data using t-SNE. *JMLR*, 9, 2579–2605.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning*, Ch. 12.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*, Ch. 14.
- McInnes, L., Healy, J., & Melville, J. (2018). UMAP: Uniform Manifold Approximation. *arXiv:1802.03426*.

Core readings: Jolliffe (PCA theory), van der Maaten & Hinton (t-SNE), McInnes et al. (UMAP)