Lecture 4: Clustering and Dimensionality Reduction

Isaiah Hull[1,2]

[1]BI Norwegian Business School
[2]CogniFrame

November 21, 2023

Unsupervised Learning

Lecture 4: Overview

1. Clustering.
2. Dimensionality Reduction.
3. Applications.

1. Clustering

Unsupervised Learning

Overview

- Based on Ng and Ma (2023).
  - https://cs229.stanford.edu/main_notes.pdf

Unsupervised Learning

Clustering Problem

- Training set: $\{x^{(1)}, \ldots, x^{(n)}\}$
- Group data into cohesive “clusters."
- No labels $y^{(i)}$: An unsupervised learning problem.

Unsupervised Learning

k-means Clustering Algorithm

1. Initialize centroids $\mu_1, \mu_2, \ldots, \mu_k$ randomly.
2. Assign cluster labels.
3. Update centroids.
4. Repeat (2) and (3) until convergence.

Unsupervised Learning

Assigning Training Examples

- For every i, set:

$$c^{(i)} := \arg\min_j \left\| x^{(i)} - \mu_j \right\|^2$$

Unsupervised Learning

Setting Cluster Centroids

- For each j, set:

$$\mu_j := \frac{\sum_{i=1}^{n} 1\left\{c^{(i)} = j\right\} x^{(i)}}{\sum_{i=1}^{n} 1\left\{c^{(i)} = j\right\}}$$

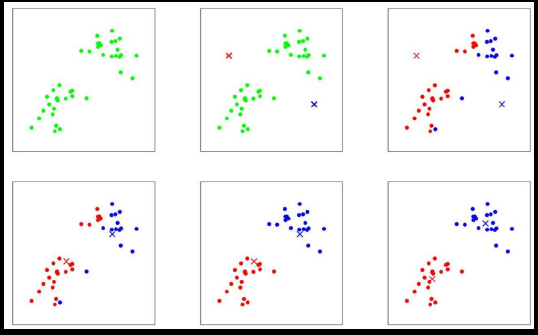
Unsupervised Learning

Understanding k-means

- k: Number of clusters.
- Centroids μ_j: Current guesses for cluster centers.
- Initialization: Randomly choose k training examples.

Unsupervised Learning

Understanding *k*-means



Source: Ng and Ma (2023).

Unsupervised Learning

Distortion Function J

$$J(c, \mu) = \sum_{i=1}^{n} \left\| x^{(i)} - \mu_{c^{(i)}} \right\|^2$$

- Measures squared distances between examples and centroids.
- k-means is coordinate descent on J.
- J must monotonically decrease.

Local Optima in k-Means

- Distortion function J is non-convex.
- k-means can be stuck in local optima.
- Solution: Run k-means multiple times and pick lowest $J(c, \mu)$.

Unsupervised Learning

k-Means: Best Practices

- Standardize features.
- Avoid highly correlated input features.
- Consider orthogonalizing features.

Unsupervised Learning

Understanding *k*-means

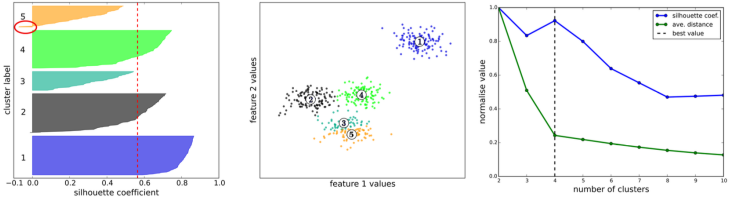


Figure 7: *k*-means cluster example. Left: Silhouette plot. Negative values (red circle) indicate the miss-assignment of observations. The vertical red line indicates the average silhouette coefficient for this clustering. Middle: Scatter plot of data of simulated data in a two dimensional feature space. Assigned cluster labels are shown for $k = 5$. Right: Silhouette and "elbow" number dependent on the number of clusters. Both methods point to an optimal number of $k = 4$. Source: Pedregosa (2011) and authors' calculations.

Source: Chakraborty and Joseph (2017).

Hierarchical Clustering

- Based on Chakraborty and Joseph (2017).
  - "Machine Learning at Central Banks"

Contrast with k-Means

- Can be thought of as a generalization of k-means clustering.
- Generates a spectrum of clusters at different resolutions, starting with the entire sample and ranging to individual examples.
- Uses entropy measures or Gini coefficient as resolution parameters.

Components of Hierarchical Clustering

1. Clustering direction: bottom-up vs. top-down.
2. Cluster formation rules.
3. Similarity measures between clusters.

Bottom-up Clustering Approach

- Starts with singletons, merging similar clusters step by step.
- More computationally feasible than top-down approach.

Top-down Clustering Approach

- Begins with all observations in one cluster.
- Splits clusters to maximize dissimilarity.
- More intensive computationally, often reliant on heuristics.

Criteria in Agglomerative Clustering

- Single linkage: Minimal distance between two points in different clusters.
- Complete linkage: Maximal distance between two points in different clusters.
- Average linkage: Average distance between points in different clusters.
- Ward criterion: Minimizes increase in within-cluster variance.

Distance Metrics in Hierarchical Clustering

- Euclidean distance: $\sqrt{\sum_{i=1}^{n}(x_i - z_i)^2}$.
- Cosine distance: $1 - \frac{x \cdot z}{\|x\| \cdot \|z\|}$.
- Manhattan distance: $\sum_{i=1}^{n} |x_i - z_i|$.

Dendrogram in Hierarchical Clustering

- Uses dendrograms for visualizing results.
- Case study example: Relationship between HCA and k-means clustering.

Unsupervised Learning

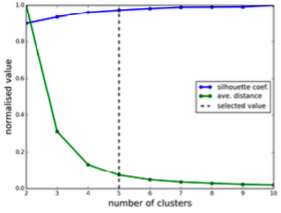
Example: Tech Start-Up Funding

name	funding / year	rounds / year	investors / year	investor score
count	32817	32817	32817	32817
mean	4.28e+06	0.55	1.03	1.22
std	1.26e+08	4.03	9.53	0.34
min	4.00e-02	0.02	0.02	1.00
25%	7.59e+04	0.07	0.10	1.00
50%	4.43e+05	0.28	0.32	1.00
75%	1.91e+06	0.63	0.84	1.40
max	2.08e+10	365.00	1095.00	2.00

Table 10: Summary statistics of constructed investment features for technology start-ups. Features are normalised using z-scores for the clustering analyses. Source: CrunchBase (TechCrunch (2015)) and authors' calculations.

Source: Chakraborty and Joseph (2017).

Example: Tech Start-Up Funding



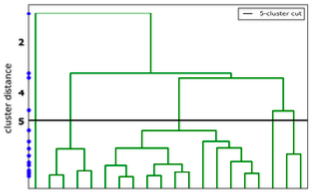


Figure 20: Comparative clustering analyses of technology funding data. Left: Elbow and silhouette coefficients for k-means clustering and k ranging from 2-10. The vertical dashed line indicates the selected number of clusters $k = 5$. Right: Dendrogram for HCA. The blue dots on the vertical axis indicate the relative values of the resolution parameter for cluster split points. The corresponding number of clusters is shown for low k-values. The horizontal line indicates the selected value $k = 5$ corresponding to the dashed line on the LHS. Source: CrunchBase (TechCrunch (2015)) and authors' calculations.

Source: Chakraborty and Joseph (2017).

Unsupervised Learning

Example: Tech Start-Up Funding

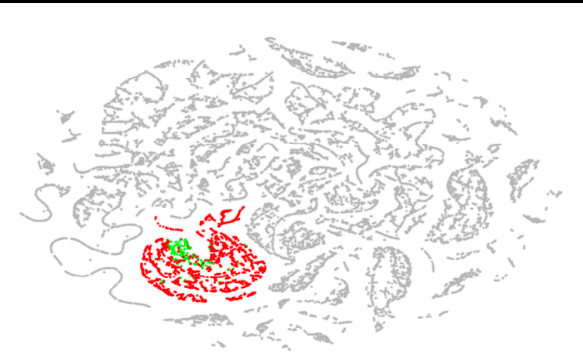


Figure 21: k-means clustering visualised in two dimensions based using t-SNE (van der Maaten and Hinton (2008)). The unicorn cluster is highlighted in red and actual unicorns are shown in green. The majority of firms in the unicorn cluster are well separated from other clusters. Axis dimensions do not have any particular meaning and are not shown. Source: CrunchBase (TechCrunch (2015)) and authors' calculations.

Source: Chakraborty and Joseph (2017).

Unsupervised Learning

Examples from de Prado (2020) "Clustering"

1. Factor investing.
2. Risk management.
3. Decomposing bond return drivers.
4. Computing p-values in multicollinear systems.

2. Dimensionality Reduction

Unsupervised Learning

High-Dimensional Problems

- ML problems often high-dimensional.
- Example: NLP with potential sequences of words.
- 1000 common words in 50-word paragraph: 10^{150} permutations.

Dimensionality Reduction in Economics and Finance

- Used when risk of overfitting or model assumptions violated.
- Techniques: PCA and FA.
- Reduces data to few factors of interest.

Unsupervised Learning

Machine Learning Methods

- PCA, PLS, and autoencoders.
- Autoencoders: “compression" and “decompression."
- Provides deep learning approach to dimensionality reduction.

Dimensionality Reduction in Economics

- Notation from Gentzkow et al. (2019) for text analysis.
- Dataset: GDP growth in 25 countries from 1961:Q2 to 2020:Q1.

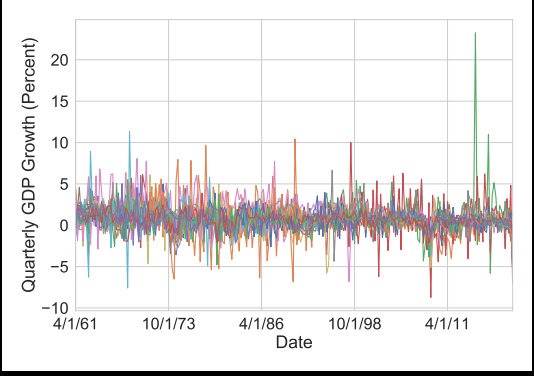
Unsupervised Learning

Purpose of the Exercises

- Extract common growth components across countries.
- PCA shows variance in growth from common components.
- Relate components to individual country series.

Unsupervised Learning

GDP Growth Dataset



Source: Hull (2021).

Note: GDP growth for 25 countries from 1961:Q2 to 2020:Q1.

Principal Component Analysis (PCA)

- Common method for dimensionality reduction.
- Maps features to k principal components.
- Components ordered by variance explained.
- First component explains largest variance.
- Components are orthogonal.

PCA Formulation

- Notation from Gentzkow et al. (2019).
- PCA as a minimization problem:

$$\min_{G,B} \text{trace}[(C - GB^T)(C - GB^T)]^T$$

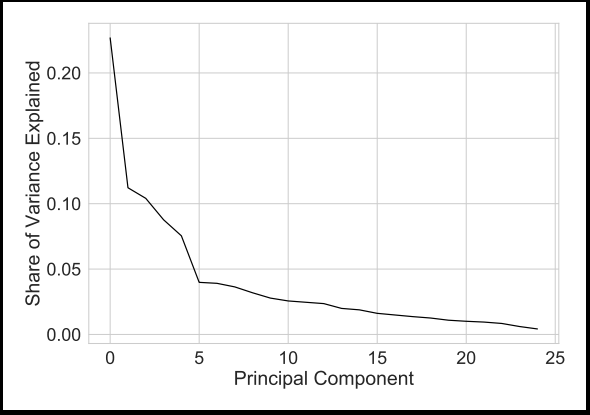
$$\text{s.t. } \text{rank}(G) = \text{rank}(B) = k$$

- C = feature matrix.
- G = principal components.
- B = association between principal component and factor.

Selecting Number of Components

- Use the “elbow method."
- Plot explained variance to find the “elbow."

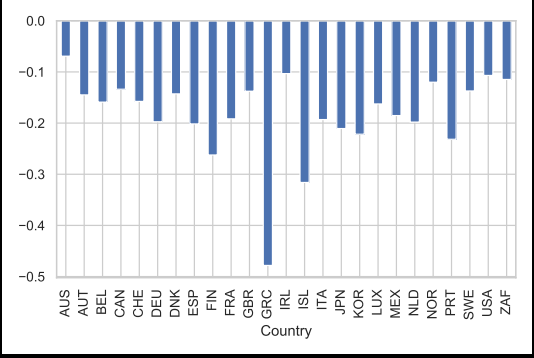
Explained Variance Share



Source: Hull (2021).

Unsupervised Learning

Strength of Association with First PC



Source: Hull (2021).

Unsupervised Learning

Applications and Context

- PCA used in broader problems.
- Example: Principal components regression (PCR).
- Two-step: PCA then include components in regression.
- Bernanke et al. (2005) used a variant for FAVAR.

Predicting Canada's GDP Growth

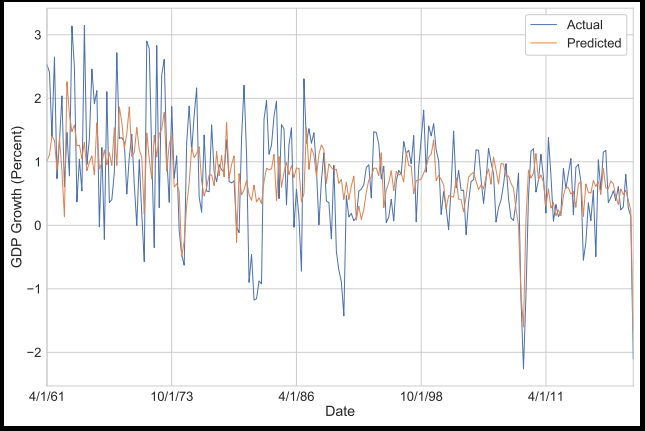
- Predict Canada's GDP growth using other countries' data.
- Use for imputation or to understand coefficient estimates.
- Principal components regression model:

$$gdp\_growth_t^{CAN} = \alpha + \beta_0 PC_{t0} + \cdots + \beta_{p-1} PC_{tp-1} + \epsilon_t$$

Model Training and Predictions

- Trained model predicts Canada's GDP growth.
- Before mid-1980s, more volatility.
- After 1980, GDP growth mostly explained by 5 factors from 24 other countries.

Actual and PCR-Predicted GDP Growth



Source: Hull (2021).

Unsupervised Learning

Implications

- Common global factors influence growth.
- Examine relationships in greater depth using B matrix.
- PCA helps in dimensionality reduction, but may weaken interpretation.

Partial Least Squares (PLS)

- PCR effectively explained Canadian GDP growth using five principal components.
- Doesn't account for relationship between C and Y in first stage.
- PLS as an alternative: uses co-movement strength between Y and feature columns of C.

PLS Steps

1. Compute:

$$\hat{Y} = \frac{\sum_j \psi_j C_j}{\sum_j \psi_j} \tag{1}$$

   - C_j is jth feature column.
   - ψ_j is univariate covariance between C_j and Y.
2. Orthogonalize Y and C with respect to $\hat{Y}$.
3. Repeat above steps for desired number of components.

Advantages of PLS

- Uses covariance between Y and C.
- Components are more suited for prediction of Y.
- Better predictive value than two-step PCA procedure.

Comparison and Variability

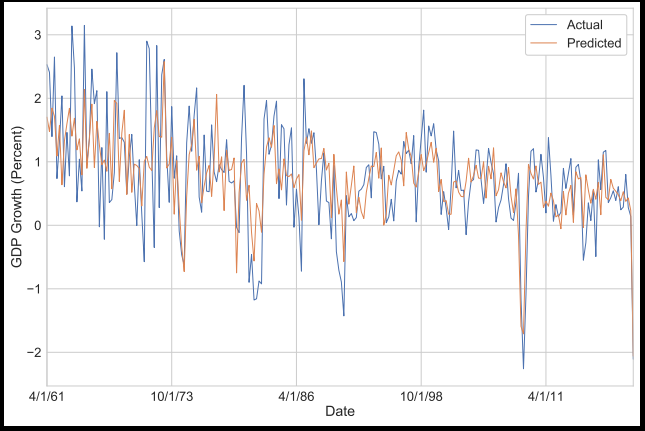
- PLS vs PCR: Both can take various forms.
- While PCR used OLS, any model could be used for capturing relationship.

Unsupervised Learning

Further Reading

- Deeper econometric theory on PLS.
  - See Kelly and Pruitt (2013) and Kelly and Pruitt (2015).
- Forecasting with PCA: Stock and Watson (2002).
- Weekly GDP during COVID-19: Lewis et al. (2020).

Actual and PCR-Predicted GDP Growth



Source: Hull (2021).

Introduction to Autoencoders

- Autoencoder: Neural network trained to predict its input values.
- Can be considered a non-linear generalization of PCA.
- Historical development: LeCun (1987), Bourlard and Kamp (1988), and Hinton and Zemel (1993).

Unsupervised Learning

Autoencoder Structure

- Consists of two functions: Encoder and Decoder.
- Encoder ($h = f(x)$): Converts inputs (x) to a latent state (h).
- Decoder ($r = g(h)$): Uses the latent state to reconstruct inputs.

Training an Autoencoder

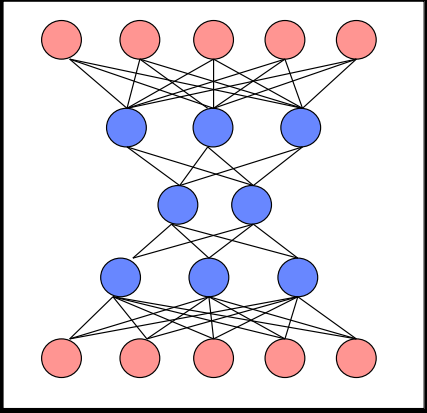
- Goal: Minimize the loss function $L(x, g(f(x)))$.
- The closer the reconstructed (r) is to the original input (x), the smaller the loss.

Unsupervised Learning

Autoencoder Architecture

- Encoder: Dense neural network for “compression."
- Decoder: Inverted neural network for “decompression."
  - Five input nodes compressed to 2 latent nodes.
  - Decoder upsamples from 2 latent nodes back to 5 output nodes.

Example Autoencoder Architecture



Source: Hull (2021).

Extended Uses of Autoencoders

- Can be used to perform dimensionality reduction.
- Other applications: Noise reduction and generative machine learning.

Noise Reduction with Autoencoders

- Applicable to both audio and images.
- Filters out noise by focusing on significant features.
- Architecture with limited nodes in latent state:
  - Compresses all information into a few numbers.
  - Reconstructs only denoised version due to data compression.

Generative Machine Learning

- Autoencoders can generate new instances of a class.
- Decoder reconstructs examples from latent state.
- New example can be created by randomly generating a latent state.
- Existing examples can be manipulated via latent state alterations.

Unsupervised Learning

Autoencoder Features

- Learn important relationships, not memorization.
- Regularization and network size are important.
- Latent state as a bottleneck.
- Latent state useful for dimensionality reduction.

Unsupervised Learning

Example: GDP Growth Data and Autoencoders

- Use autoencoder for GDP growth data.
- Encoder and decoder models share weights.
- Set number of nodes in latent state (latentNodes) to five.
- Comparable to a five-factor PCR model.

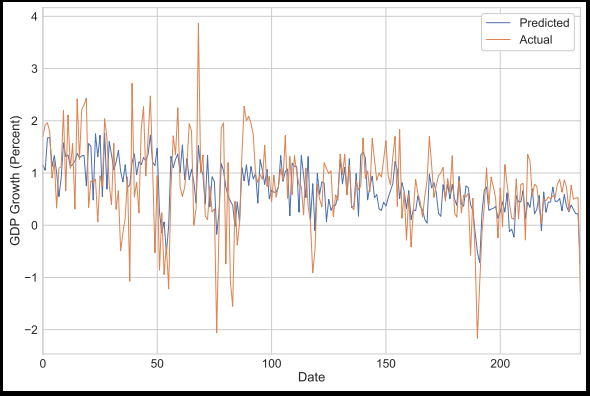
Unique Aspects of Autoencoders in our Model

- Features and target are the same.
- Separate encoder and decoder models, yet part of a larger model.
- Simple architecture: latent state with five nodes.
- Trained on 24 GDP growth series with 236 quarters each.
- A total of only 269 parameters.

Reproducing the GDP Growth Series

- Autoencoder attempts to reproduce GDP growth series.
- Bottleneck layer forces noise reduction.
- Predicted series has lower variance due to noise reduction.
- Use encoder's predict method to recover latent state time series.

Reconstructed U.S. Growth using Autoencoder



Source: Hull (2021).

Regression on Latent State Time Series

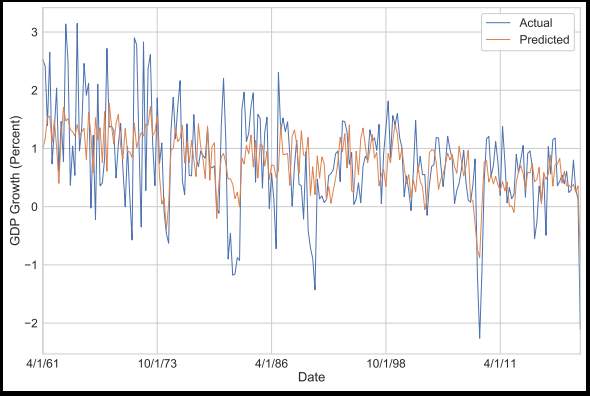
- Use latent states to predict Canada's GDP growth.
- Similar linear regression approach as with PCR.
- Performance matches PLS.

Unsupervised Learning

Alternative Approaches

- Modify autoencoder's architecture.
- Use a different model for the second step.
- With TensorFlow, connect model directly to autoencoder.
- Yield a PLS-type generalization by predicting Y with five latent features.

Actual and OLS-Predicted Growth using Autoencoder



Source: Hull (2021).

Summary of Dimensionality Reduction

- Common strategy in both economics and machine learning.
- Often used when supervised learning tasks are infeasible with the given feature set.
- Methods: Principal components analysis, latent states from autoencoders.

Summary of Dimensionality Reduction

- PCR performs well, but first step does not consider dependent variable.
- PLS showed slight improvements by exploiting comovement with dependent variable.

Summary of Dimensionality Reduction

- Autoencoder consists of encoder and decoder networks.
- Trained to reproduce its inputs.
- Encoder produces a latent state: compressed representation.
- Regressions using autoencoder latent states compared favorably to PLS.
- Potential: Joint training with the predictive model.

3. Applications

Introduction to TensorFlow

Tutorials

- Dimensionality Reduction
- Cluster Analysis in Python - DataCamp

References I

Bernanke, B.S., J. Boivin, and P. Eliasz (2005) "Measuring the Effects of Monetary Policy: A Factor-Augmented Vector Autoregressive (FAVAR) Approach," *The Quarterly Journal of Economics*, 120 (1), 387–422.

Bourlard, H. and Y. Kamp (1988) "Auto-association by multilayer perceptrons and singular value decomposition," *Biological Cybernetics*, 59, 291–294.

Chakraborty, Chiranjit and Andreas Joseph (2017) "Machine learning at central banks," Bank of England working papers 674, Bank of England, https://ideas.repec.org/p/boe/boeewp/0674.html.

Hinton, G.E. and R.S. Zemel (1993) "Autoencoders, minimum description length, and Helmholtz free energy," in *NIPS'1993*.

References II

Hull, Isaiah (2021) *Machine Learning for Economics and Finance in TensorFlow 2*: Apress, 10.1007/978-1-4842-6373-0.

Kelly, B. and S. Pruitt (2013) "Market Expectations in the Cross-Section of Present Values," *Journal of Finance*, 68 (5), 1721–1756.

——— (2015) "The Three-Pass Regression Filter: A New Approach to Forecasting Using Many Predictors," *Journal of Econometrics*, 186 (2), 294–316.

LeCun, Y. (1987) *Modèles connexionistes de l'apprentissage* Ph.D. dissertation, Université de Paris VI.

Lewis, D., K. Mertens, and J. Stock (2020) "U.S. Economic Activity during the Early Weeks of the SARS-Cov-2 Outbreak," Staff Reports 920, Federal Reserve Bank of New York.

References III

Ng, Andrew and Tengyu Ma (2023) "CS229 Lecture Notes," June.

Stock, J.H. and M.W. Watson (2002) "Forecasting Using Principal Components from a Large Number of Predictors," *Journal of the American Statistical Association*, 97 (460), 1167–1179.