# Comprehensive Notes on Unsupervised Learning

## Introduction to Unsupervised Learning

Unsupervised learning represents a distinct paradigm within machine learning, where the goal is to uncover inherent structure within data without the guidance of labeled responses. Unlike supervised learning methods that map inputs to known outputs, unsupervised learning algorithms work with observations X₁, X₂, ..., Xₙ without corresponding response variables Y.

The central objective is to discover intrinsic patterns, relationships, and properties within the dataset. Unsupervised learning frequently serves as an exploratory data analysis tool, enabling researchers to gain insights into data organization before proceeding with more directed analyses.

Key characteristics of unsupervised learning include:

* No response variables for validation
* Results tend to be more subjective and interpretative
* Lack of standard validation mechanisms
* Focus on pattern discovery rather than prediction
* Often serves as a preprocessing step for supervised methods

Two predominant techniques in unsupervised learning are Principal Component Analysis (PCA) and Clustering methods.

## Principal Component Analysis (PCA)

### Conceptual Framework

Principal Component Analysis is a dimensionality reduction technique that transforms potentially correlated variables into a set of linearly uncorrelated variables called principal components. These components are ordered by the amount of variance they explain in the original dataset.

PCA can be interpreted through multiple lenses:

1. **Variance Maximization**: Finding directions in feature space along which data exhibits maximum variability
2. **Approximation**: Determining lines and subspaces closest to the data points in terms of Euclidean distance
3. **Data Compression**: Creating a lower-dimensional representation that preserves maximum information

### Mathematical Formulation

The first principal component of a set of features X₁, X₂, ..., Xₚ is the normalized linear combination:

Z₁ = φ₁₁X₁ + φ₂₁X₂ + ... + φₚ₁Xₚ

subject to the normalization constraint:

∑ⱼ₌₁ᵖ φⱼ₁² = 1

where φⱼ₁ are the loadings of the first principal component, collectively forming the principal component loading vector φ₁ = (φ₁₁, φ₂₁, ..., φₚ₁).

This normalization constraint prevents arbitrary scaling that could artificially inflate variance.

### Computation Process

To compute PCA for an n×p data matrix X:

1. **Center the variables** to have zero mean
2. **Solve the optimization problem** that maximizes the sample variance:

Maximize\_{φ₁₁,...,φₚ₁} {(1/n)∑ᵢ₌₁ⁿ(∑ⱼ₌₁ᵖφⱼ₁xᵢⱼ)²}

subject to:

∑ⱼ₌₁ᵖ φⱼ₁² = 1

This can be rewritten as:

(1/n)∑ᵢ₌₁ⁿzᵢ₁²

where zᵢ₁ represents the scores of the first principal component.

1. **Eigendecomposition** is typically used to solve this optimization problem

### Subsequent Principal Components

The second principal component Z₂ is the linear combination that: - Maximizes variance among all possible linear combinations - Is uncorrelated with Z₁ - Has normalized loadings

For observation i, the second component score is:

zᵢ₂ = φ₁₂xᵢ₁ + φ₂₂xᵢ₂ + ... + φₚ₂xᵢₚ

The constraint that Z₂ be uncorrelated with Z₁ is mathematically equivalent to requiring the loading vectors φ₁ and φ₂ to be orthogonal.

This process continues for all possible principal components, which can number up to Min(n-1, p).

### Geometric Interpretation

PCA has several geometric interpretations:

1. **Directional Variance**: The first principal component defines the direction in feature space with maximum data variance
2. **Projection**: Principal component scores represent the projection of data points onto these directions
3. **Minimum Distance**: The first M principal components define an M-dimensional hyperplane closest to the data points in terms of squared Euclidean distance

The projection interpretation means that the original data can be approximated as:

xᵢⱼ ≈ ∑ₘ₌₁ᴹ zᵢₘφⱼₘ

When M = Min(n-1, p), this approximation becomes exact:

xᵢⱼ = ∑ₘ₌₁ᴹ zᵢₘφⱼₘ

### Variance Explained

The portion of variance explained by principal components provides a quantitative measure of their importance:

1. **Total variance** in the dataset (assuming centered variables):

∑ⱼ₌₁ᵖ Var(Xⱼ) = ∑ⱼ₌₁ᵖ (1/n)∑ᵢ₌₁ⁿ xᵢⱼ²

1. **Variance explained** by the mth principal component:

(1/n)∑ᵢ₌₁ⁿ zᵢₘ² = (1/n)∑ᵢ₌₁ⁿ(∑ⱼ₌₁ᵖ φⱼₘxᵢⱼ)²

1. **Portion of variance explained** by the mth principal component:

∑ᵢ₌₁ⁿ(∑ⱼ₌₁ᵖ φⱼₘxᵢⱼ)² / ∑ⱼ₌₁ᵖ∑ᵢ₌₁ⁿ xᵢⱼ²

The cumulative portion of variance explained by the first M components is the sum of their individual portions. All principal components together explain 100% of the variance.

### Practical Considerations

Several important practical aspects of PCA include:

1. **Standardization**: Variables are typically standardized before PCA since results depend on scale
2. **Selecting Number of Components**: Several approaches exist:

* Scree plots (looking for "elbow" points)
* Cumulative variance explained threshold (e.g., 80-90%)
* Cross-validation in supervised contexts

1. **Uniqueness**: Principal components are unique (aside from possible sign changes)
2. **Application**: PCA can be used as a preprocessing step for other methods to reduce noise

## Clustering Methods

Clustering algorithms partition data into subgroups where observations within each group share similar characteristics while differing from observations in other groups. The definition of similarity is context-dependent and influences the clustering approach.

### K-Means Clustering

K-means clustering partitions data into K distinct, non-overlapping clusters, where K is specified in advance.

#### Mathematical Formulation

K-means operates under two key constraints: 1. Each observation belongs to at least one cluster: C₁ ∪ C₂ ∪ ... ∪ Cₖ = {1, ..., n} 2. Clusters are non-overlapping: Cₖ ∩ Cₖ' = ∅ for all k ≠ k'

The objective is to minimize the total within-cluster variation:

∑ₖ₌₁ᴷ W(Cₖ)

where W(Cₖ) measures within-cluster variation.

Using squared Euclidean distance as the measure yields:

W(Cₖ) = (1/|Cₖ|)∑ᵢ,ᵢ'∈Cₖ ∑ⱼ₌₁ᵖ(xᵢⱼ - xᵢ'ⱼ)²

where |Cₖ| denotes the number of observations in cluster k.

The full optimization problem becomes:

Minimize\_{C₁,...,Cₖ} {∑ₖ₌₁ᴷ (1/|Cₖ|)∑ᵢ,ᵢ'∈Cₖ ∑ⱼ₌₁ᵖ(xᵢⱼ - xᵢ'ⱼ)²}

#### Algorithm

Since finding the global optimum is computationally infeasible for non-trivial datasets, K-means uses an iterative algorithm:

1. **Initialize**: Randomly assign observations to clusters 1 through K
2. **Iterate until convergence**:
3. Compute the centroid of each cluster (mean of all observations in the cluster)
4. Reassign each observation to the cluster with the nearest centroid

This algorithm guarantees convergence to a local optimum by consistently decreasing the objective function with each iteration.

#### Practical Considerations

1. **Multiple Initializations**: Since K-means finds local optima, multiple runs with different random starts are recommended
2. **Selecting K**: Various methods exist:

* Domain knowledge
* Elbow method (plotting within-cluster sum of squares vs. K)
* Silhouette analysis
* Gap statistic

1. **Limitations**:

* Requires pre-specifying K
* Can be sensitive to outliers
* Assumes spherical cluster shapes
* Can converge to suboptimal solutions

### Hierarchical Clustering

Hierarchical clustering creates a nested hierarchy of clusters without requiring pre-specification of cluster number. It produces a tree-based representation called a dendrogram.

#### Types of Hierarchical Clustering

1. **Agglomerative (Bottom-up)**: Start with each observation as its own cluster and progressively merge
2. **Divisive (Top-down)**: Start with all observations in one cluster and progressively divide

Agglomerative clustering is more common and proceeds as follows:

1. Treat each observation as its own cluster (n clusters)
2. Compute pairwise dissimilarities between all clusters
3. Merge the two most similar clusters
4. Recalculate dissimilarities between the new cluster and all remaining clusters
5. Repeat steps 3-4 until all observations belong to a single cluster

#### Dendrogram Interpretation

A dendrogram visually represents the hierarchical clustering process: - Height corresponds to dissimilarity between merged clusters - Earlier fusions (lower in the tree) indicate greater similarity - Horizontal proximity has no interpretive value - Horizontal cuts produce distinct clusterings

Clusters can be extracted by making a horizontal cut across the dendrogram. The height of this cut controls the number of resulting clusters, analogous to K in K-means clustering.

#### Linkage Methods

Linkage defines how dissimilarity is measured between groups of observations:

1. **Complete Linkage**: Maximum pairwise dissimilarity between observations in different clusters

* Formula: max{d(x, y) : x ∈ A, y ∈ B}
* Tends to produce more balanced clusters
* Sensitive to outliers

1. **Single Linkage**: Minimum pairwise dissimilarity between observations in different clusters

* Formula: min{d(x, y) : x ∈ A, y ∈ B}
* Can produce extended, chain-like clusters
* Less sensitive to outliers

1. **Average Linkage**: Mean pairwise dissimilarity between observations in different clusters

* Formula: (1/|A||B|)∑ₓ∈ₐ∑ᵧ∈ᵦ d(x, y)
* Compromise between complete and single linkage
* Generally produces balanced clusters

1. **Centroid Linkage**: Dissimilarity between cluster centroids

* Formula: d(centroid(A), centroid(B))
* Can produce inversions in the dendrogram
* Popular in genomics applications

#### Dissimilarity Measures

Various measures can quantify dissimilarity between observations:

1. **Euclidean Distance**: sqrt(∑ⱼ₌₁ᵖ(xᵢⱼ - xᵢ'ⱼ)²)

* Sensitive to magnitudes
* Affected by scaling

1. **Correlation-Based Distance**: 1 - cor(xᵢ, xᵢ')

* Focuses on pattern shapes rather than magnitudes
* Less sensitive to scaling
* Useful when patterns are more important than absolute values

1. **Manhattan Distance**: ∑ⱼ₌₁ᵖ|xᵢⱼ - xᵢ'ⱼ|

* Less sensitive to outliers than Euclidean distance
* Appropriate for grid-like feature spaces

1. **Gower Distance**: Weighted combination of distances for mixed data types

* Handles categorical, ordinal, and continuous variables together
* Allows different weights for different features

### Advanced Clustering Considerations

Several advanced topics in clustering deserve attention:

1. **Robust Clustering**: Techniques that are less sensitive to outliers or noise

* Trimmed K-means
* Density-based methods (DBSCAN)

1. **Partial Clustering**: Methods that don't force all observations into clusters

* DBSCAN
* Mixture models
* Soft K-means (fuzzy clustering)

1. **Statistical Significance**: Methods to assess cluster validity

* Silhouette coefficients
* Gap statistic
* Bootstrap approaches

1. **Stability Assessment**: Techniques to evaluate clustering robustness

* Subsample and re-cluster
* Add noise and re-cluster
* Compare clusters across different algorithms

## Decision Making in Unsupervised Learning

The application of unsupervised learning requires several important decisions:

1. **Data Preprocessing**:

* Centering and scaling variables
* Handling missing values
* Transforming skewed distributions

1. **Method Selection**:

* PCA for linear dimensionality reduction
* t-SNE or UMAP for nonlinear dimensionality reduction
* K-means for spherical clusters
* Hierarchical clustering for nested structure
* DBSCAN for arbitrary-shaped clusters and outlier detection

1. **Parameter Tuning**:

* For K-means: number of clusters K
* For hierarchical clustering: linkage method, dissimilarity measure, dendrogram cut height
* For PCA: number of components to retain

1. **Validation**:

* Internal validation measures (silhouette, Davies-Bouldin)
* Stability assessment
* Domain expert evaluation

1. **Interpretation**:

* Connect clusters to domain-specific meaning
* Visualize PCA loadings to understand component meanings
* Use additional information to validate interpretations

## Extended Applications

Unsupervised learning extends beyond basic PCA and clustering:

1. **Feature Engineering**: Using PCA components as inputs to supervised models
2. **Anomaly Detection**: Identifying outliers that don't fit established patterns
3. **Market Segmentation**: Finding customer groups with similar preferences
4. **Document Clustering**: Organizing text documents by topic similarity
5. **Image Compression**: Representing images with fewer dimensions
6. **Network Analysis**: Detecting communities in social networks
7. **Recommender Systems**: Grouping users with similar preferences

## Recent Advancements

Contemporary research in unsupervised learning has expanded the field:

1. **Autoencoders**: Neural network-based nonlinear dimensionality reduction
2. **Variational Autoencoders (VAEs)**: Probabilistic autoencoders for generative modeling
3. **Self-Supervised Learning**: Learning representations without labels through pretext tasks
4. **Contrastive Learning**: Learning representations by comparing similar and dissimilar samples
5. **Spectral Clustering**: Graph-based clustering using eigenvectors of similarity matrices
6. **Deep Clustering**: Integrating deep learning with clustering objectives

## Conclusion

Unsupervised learning provides powerful tools for exploring and understanding data structure without labeled responses. By revealing inherent patterns and reducing dimensionality, these methods facilitate both data comprehension and subsequent analysis. PCA offers a linear approach to dimension reduction, while various clustering techniques identify natural groupings in data. The effectiveness of unsupervised learning depends on appropriate choices of methods, parameters, and validation approaches tailored to the specific application context.