Análisis Comparativo de Algoritmos de Clustering: K-means vs Self-Organizing Maps (SOM)

Taller No. 1 - Aprendizaje No Supervisado

Programa de Posgrado en Big Data
Universidad [Nombre de la Universidad]

27 de septiembre de 2025

Resumen

Este informe presenta un análisis comparativo exhaustivo entre los algoritmos de clustering K-means y Self-Organizing Maps (SOM) aplicados a dos conjuntos de datos con diferentes niveles de complejidad. Se implementó una metodología semi-supervisada con validación cruzada para evaluar la efectividad de ambos algoritmos en tareas de clasificación. Los resultados demuestran que SOM supera significativamente a K-means en datasets complejos multi-clase (45.79 % vs 33.10 % de accuracy), mientras que K-means mantiene competitividad en problemas simples con clusters bien definidos. El análisis incluye preprocesamiento avanzado con balanceamiento híbrido, optimización de hiperparámetros, y evaluación mediante múltiples métricas cuantitativas.

Índice

1. Introducción

El clustering es una técnica fundamental en el aprendizaje no supervisado que busca agrupar datos similares sin conocimiento previo de las etiquetas. En este trabajo se comparan dos algoritmos representativos: K-means, basado en centroides, y Self-Organizing Maps (SOM), basado en redes neuronales competitivas.

1.1. Objetivos

1.1.1. Objetivo General

Evaluar y comparar el rendimiento de los algoritmos K-means y SOM en diferentes tipos de datasets, utilizando una metodología semi-supervisada con validación cruzada.

1.1.2. Objetivos Específicos

- Identificar configuraciones óptimas para ambos algoritmos en datasets simples y complejos
- Implementar una metodología semi-supervisada para evaluación de clustering
- Analizar la efectividad del balanceamiento de clases en datasets desequilibrados
- Comparar algoritmos mediante múltiples métricas cuantitativas
- Proporcionar recomendaciones prácticas para la selección de algoritmos

1.2. Hipótesis

Se plantea que SOM demostrará superioridad en datasets complejos multi-dimensionales debido a su capacidad de preservar la topología del espacio de características, mientras que K-means será más efectivo en datasets simples con clusters esféricos bien definidos.

2. Marco Teórico

2.1. K-means

K-means es un algoritmo de particionamiento que busca dividir los datos en k clusters minimizando la suma de distancias cuadráticas intra-cluster:

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2 \tag{1}$$

donde μ_i es el centroide del cluster C_i.

2.2. Self-Organizing Maps (SOM)

SOM es una red neuronal no supervisada que proyecta datos de alta dimensión en un mapa topológico de menor dimensión, típicamente bidimensional. La función de vecindad se define como:

$$h_{c,i}(t) = \alpha(t) \cdot \exp\left(-\frac{||r_c - r_i||^2}{2\sigma^2(t)}\right) \tag{2}$$

donde $\alpha(t)$ es la tasa de aprendizaje y $\sigma(t)$ el radio de vecindad.

3. Metodología

3.1. Datasets Utilizados

3.1.1. Dataset 1: data_clusters.mat

- **Características**: 136 muestras, 2 dimensiones
- **Tipo**: Dataset sintético con clusters bien definidos
- **Propósito**: Evaluación en problema simple

3.1.2. Dataset 2: Avila UCI

- **Características**: 20,867 muestras originales, 10 dimensiones, 8 clases
- **Tipo**: Dataset real de clasificación paleográfica
- **Propósito**: Evaluación en problema complejo multi-clase
- **Descripción**: Clasificación de copistas de manuscritos del siglo XII

3.2. Preprocesamiento

3.2.1. Balanceamiento Híbrido (Dataset Avila)

Se implementó una estrategia en dos fases para corregir el desequilibrio de clases (ratio original 12.16:1):

1. **Fase 1 - Undersampling**: Reducción de la clase mayoritaria (A) de 8,572 a 4,500 muestras
2. **Fase 2 - SMOTE**: Generación sintética para elevar clases minoritarias a 2,500 muestras cada una

Resultado: 23,423 muestras balanceadas con ratio 1.80:1 (mejora de 6.8x).

3.2.2. Normalización

- **K-means**: StandardScaler (media=0, desviación=1)
- **SOM**: MinMaxScaler (rango [0,1]) posterior a StandardScaler

3.3. División de Datos

Se implementó división estratificada 70/15/15:

- **70 % Entrenamiento**: Clustering no supervisado
- **15 % Etiquetado**: Mapeo cluster → clase
- **15 % Validación**: Evaluación final

3.4. Configuraciones Evaluadas

3.4.1. K-means

- **Dataset simple**: K ∈ [2, 10] (9 configuraciones)
- **Dataset Avila**: K ∈ [2, 20] (19 configuraciones, 5 ejecuciones c/u)
- **Métricas**: Método del codo, coeficiente de silhouette

3.4.2. SOM

- **Dataset simple**: Rejillas 3×3 hasta 8×8 (15 configuraciones)
- **Dataset Avila**: Rejillas 3×3 hasta 10×10 (22 configuraciones)
- **Métricas**: Error de cuantización, eficiencia neuronal

4. Resultados

4.1. Dataset Simple (data_clusters.mat)

4.1.1. Configuraciones Óptimas

Cuadro 1: Resultados Óptimos - Dataset Simple

Algoritmo	Configuración	Métrica Principal
K-means	K=6	Silhouette = 0.7396
SOM	7×7 (35 activas)	Error cuant. = 0.0241

4.1.2. Análisis Comparativo

K-means demostró ligera superioridad en el dataset simple debido a:

- Clusters esféricos bien definidos
- Baja dimensionalidad (2D)
- Ausencia de ruido significativo

4.2. Dataset Complejo (Avila UCI)

4.2.1. Configuraciones Óptimas

Cuadro 2: Resultados Óptimos - Dataset Avila

Algoritmo	Configuración	Métrica Principal
K-means	K=10	Silhouette = 0.2176 ± 0.0283
SOM	8×8 (63 activas)	Error cuant. = 0.1068

4.2.2. Resultados de Validación Cruzada

Cuadro 3: Rendimiento Semi-supervisado - Validación Cruzada 5-fold

Algoritmo	Accuracy CV	Accuracy Final
K-means	0.2955 ± 0.0304	33.10 %
SOM	0.4214 ± 0.0102	45.79 %
Ventaja SOM	**+42.6 %**	**+38.2 %**

4.3. Análisis de Calidad de Etiquetado

Se evaluó la calidad del proceso de etiquetado midiendo la pureza de cada cluster/-neurona:

Cuadro 4: Calidad del Etiquetado

Métrica	K-means	SOM	Ganador
Pureza Promedio	39.7 %	50.5 %	SOM
Pureza Mediana	36.2 %	40.7 %	SOM
Pureza Mínima	20.1 %	21.3 %	SOM
Pureza Máxima	91.0 %	100.0 %	SOM
Desv. Std. Pureza	20.1 %	23.7 %	K-means
Métricas Ganadas	**1/5**	**4/5**	**SOM**

4.4. Matrices de Confusión

Las matrices de confusión revelan patrones importantes en el rendimiento:

4.4.1. K-means

- Fuerte sesgo hacia clasificación como clase F
- Recall prácticamente nulo para clases D, E, G, H
- Mejor rendimiento solo en clase I (recall: 69 %)

4.4.2. SOM

- Distribución más equilibrada entre clases
- Mejor recall promedio (45 % vs 31 %)
- Rendimiento superior en 6 de 8 clases

4.5. Análisis de Eficiencia Computacional

Cuadro 5: Comparación de Eficiencia

Aspecto	K-means	SOM
Tiempo entrenamiento	Rápido	Moderado
Escalabilidad	Excelente	Buena
Memoria requerida	Baja	Moderada
Interpretabilidad	Alta	Media
Robustez a ruido	Media	Alta

5. Discusión

5.1. ¿Por qué SOM Superó a K-means en el Dataset Complejo?

5.1.1. Preservación de Topología

SOM mantiene las relaciones de vecindad del espacio original, crucial para datos paleográficos donde características similares deben agruparse próximamente.

5.1.2. Mejor Representación

Las 63 neuronas activas del SOM 8×8 capturan patrones más sutiles que los 10 centroides de K-means, permitiendo una segmentación más granular.

5.1.3. Flexibilidad Estructural

La rejilla bidimensional permite representar distribuciones complejas y no esféricas, comunes en datos reales de alta dimensión.

5.1.4. Menor Sensibilidad

SOM es menos afectado por outliers y ruido debido a su mecanismo de aprendizaje competitivo y función de vecindad.

5.2. Efectividad del Balanceamiento

El balanceamiento híbrido demostró ser crucial:

- Mejora del ratio de 12.16:1 a 1.80:1 (6.8× mejor)
- Reducción de desviación estándar entre clases (3.3× menor)

- Validación cruzada más estable
- Matrices de confusión más distribuidas

5.3. Implicaciones para Clasificación Paleográfica

En el contexto de clasificación de copistas manuscritos:

- **SOM** ofrece mejor interpretabilidad topológica
- Permite identificar estilos de escritura similares mediante vecindad
- La preservación de estructura es crucial para análisis paleográfico
- La granularidad de 63 neuronas vs 10 clusters mejora discriminación

6. Conclusiones

6.1. Conclusiones Principales

1. **Superioridad contextual**: SOM demuestra superioridad significativa en datasets complejos multi-clase (45.79 % vs 33.10 %), mientras K-means mantiene competitividad en problemas simples.
2. **Efectividad del balanceamiento**: La estrategia híbrida (undersampling + SMOTE) es crucial para datasets desequilibrados, mejorando el rendimiento de ambos algoritmos.
3. **Calidad de etiquetado**: SOM genera clusters con mayor pureza promedio (50.5 % vs 39.7 %) y mejor distribución de clases.
4. **Metodología semi-supervisada**: La división 70/15/15 con validación cruzada proporciona evaluación robusta y metodológicamente sólida.

6.2. Recomendaciones Prácticas

6.2.1. Usar K-means cuando:

- Datos tienen clusters aproximadamente esféricos
- Se requiere interpretabilidad simple (centroides)
- Eficiencia computacional es prioritaria
- Dimensionalidad baja-media (¡10 características)

6.2.2. Usar SOM cuando:

- Datos tienen estructura topológica compleja
- Se requiere preservación de vecindad
- Visualización de alta dimensionalidad es importante
- Se necesita detectar patrones sutiles

6.3. Configuraciones Recomendadas

Cuadro 6: Configuraciones Recomendadas por Tipo de Problema

Tipo de Problema	K-means	SOM
Dataset Simple (2D)	K=6	7×7
Dataset Complejo (¿5D)	K=10	8×8
Clasificación Paleográfica	No recomendado	8×8

7. Trabajo Futuro

7.1. Algoritmos Avanzados

- Evaluación de DBSCAN y Gaussian Mixture Models
- Implementación de autoencoders para clustering
- Clustering ensemble combinando múltiples algoritmos

7.2. Optimización

- Búsqueda automática de hiperparámetros con Optuna
- Paralelización para datasets masivos
- Implementación en GPU para SOM

7.3. Aplicaciones

- Extensión a otros dominios paleográficos
- Aplicación en clasificación de documentos históricos
- Desarrollo de sistema interactivo de análisis

8. Referencias

1. Kohonen, T. (2001). *Self-Organizing Maps*. Springer-Verlag.
2. MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1, 281-297.
3. Chawla, N. V., et al. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16, 321-357.
4. Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20, 53-65.

5. De Stefano, C., et al. (2018). Reliable writer identification in medieval manuscripts through page layout features: The .Avila"Bible case. *Engineering Applications of Artificial Intelligence*, 72, 99-110.

A. Código Principal

Listing 1: Implementación K-means Optimizado

```
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

# Configuracion K-means
k_range = range(2, 21)
best_silhouette = -1
best_k = 2

for k in k_range:
    kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
    cluster_labels = kmeans.fit_predict(X_train_scaled)
    silhouette_avg = silhouette_score(X_train_scaled, cluster_labels)

    if silhouette_avg > best_silhouette:
        best_silhouette = silhouette_avg
        best_k = k
```

Listing 2: Implementación SOM Optimizado

```
from minisom import MiniSom

# Configuracion SOM
som_configs = []
for x in range(3, 11):
    for y in range(3, 11):
        som_configs.append({'x': x, 'y': y})

best_som = None
best_score = float('inf')

for config in som_configs:
    som = MiniSom(config['x'], config['y'], X_train_scaled.shape[1],
                  sigma=1.0, learning_rate=0.5, random_seed=42)
    som.train(X_train_scaled, 1000)

    # Calcular error de cuantizacion
    quantization_error = som.quantization_error(X_train_scaled)

    if quantization_error < best_score:
        best_score = quantization_error
        best_som = som
```

B. Métricas Detalladas

B.1. Resultados Completos Dataset Avila

Cuadro 7: Resultados Detallados por Clase - Dataset Avila

Clase	**K-means**			**SOM**		
	Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
A	0.27	0.36	0.31	0.37	0.49	0.42
D	0.00	0.00	0.00	0.40	0.23	0.29
E	0.00	0.00	0.00	0.37	0.35	0.36
F	0.25	0.60	0.35	0.37	0.53	0.43
G	0.00	0.00	0.00	0.35	0.18	0.24
H	0.20	0.17	0.18	0.35	0.26	0.30
I	0.89	0.69	0.77	0.85	0.78	0.82
X	0.43	0.67	0.52	0.80	0.77	0.78
Promedio	**0.25**	**0.31**	**0.27**	**0.48**	**0.45**	**0.45**