

LEARNING VECTOR QUANTIZATION (LVQ) PARA CLASIFICACIÓN DE ACTIVIDADES EN APLICACIONES MÓVILES

Alvaro Anzuetto Ríos¹, Nohemi Barranco Alavez¹

Instituto Politécnico Nacional

¹Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas (UPIITA)

aanzuetor@ipn.mx, nohemib2702@gmail.com

Boletín No. 106, 1o. de enero de 2025

Resumen

Este trabajo presenta una implementación del algoritmo Learning Vector Quantization (LVQ) aplicado a la clasificación de actividades de usuarios en una aplicación móvil. El sistema desarrollado categoriza las interacciones en cuatro tipos: navegación, interacción social, compras y consumo de contenidos, utilizando métricas como el tiempo de sesión, número de clics, tipo de contenido accedido y cantidad de datos transferidos. La metodología implementa cuatro neuronas competitivas, cada una representando un prototipo de actividad, que se ajustan durante el entrenamiento mediante el cálculo de distancias euclidianas y la actualización de pesos. Los resultados muestran que el sistema es capaz de clasificar nuevas actividades basándose en la similitud con los prototipos aprendidos, permitiendo una categorización del comportamiento del usuario. Esta implementación demuestra la aplicabilidad del LVQ en el contexto de análisis de comportamiento de usuarios en aplicaciones móviles.

Palabras Clave: Learning Vector Quantization, clasificación de actividades, aplicaciones móviles, redes neuronales, análisis de comportamiento.

1. Introducción

El análisis del comportamiento de usuarios en aplicaciones móviles representa un área de creciente interés en el campo de la inteligencia artificial y el aprendizaje automático. Learning Vector Quantization (LVQ), introducido por Kohonen (1990), ha demostrado ser una herramienta efectiva para la clasificación de patrones en diversos contextos. Este algoritmo, basado en el aprendizaje competitivo, permite la categorización de datos mediante la adaptación de vectores prototipo que representan diferentes clases.

La clasificación de actividades de usuarios en entornos móviles requiere técnicas robustas que puedan manejar múltiples dimensiones de datos y adaptarse a patrones cambiantes. El LVQ ofrece ventajas significativas en este contexto debido a su capacidad de aprendizaje supervisado y su eficiencia computacional (Nova & Estévez, 2014). Los métodos de cuantización vectorial han evolucionado para incluir variantes que mejoran la precisión y la velocidad de convergencia (Zhang & Wu, 2017).

Wu et al., (2016) demostraron la efectividad del LVQ en aplicaciones móviles, destacando su capacidad para procesar datos en tiempo real y adaptarse a nuevos patrones de comportamiento. Esta característica es particularmente relevante en el contexto actual, donde la comprensión del comportamiento del usuario es crucial para la personalización de servicios y la mejora de la experiencia de usuario.

2. Metodología

En este trabajo se ha implementado y evaluado un modelo de Learning Vector Quantization (LVQ) orientado a la clasificación de datos en cuatro categorías distintas: Navegación, Interacción Social, Compras y Consumo de Contenidos. Se describen en detalle el algoritmo LVQ, el conjunto de datos utilizado, los procedimientos de preprocesamiento, la estructura de la red neuronal, el proceso de entrenamiento y las técnicas de validación aplicadas. La metodología adoptada se fundamenta en trabajos previos y se ajusta a las mejores prácticas en el ámbito del aprendizaje supervisado y las redes neuronales [Kohonen, 1995;

Haykin, 1999].

2.1 Descripción de Learning Vector Quantization (LVQ)

El Learning Vector Quantization (LVQ) es un algoritmo de aprendizaje supervisado propuesto inicialmente por Teuvo Kohonen en 1986 [Kohonen, 1986]. LVQ es una variante de las redes neuronales de tipo competitivo y se basa en la consideración de vectores prototipo que representan cada una de las clases en el espacio de características. El algoritmo busca ajustar estos prototipos durante el entrenamiento para que reflejen las características de los datos asociados a cada clase.

El LVQ es especialmente útil en tareas de clasificación donde las fronteras de decisión no son lineales y se requiere una interpretación intuitiva de cómo se agrupan los datos [Kohonen, 1995]. Para este trabajo, el LVQ se aplica para clasificar patrones de comportamiento en línea de usuarios, identificando su pertenencia a una de las cuatro categorías definidas. Esto es relevante en áreas como la personalización de contenido, el análisis de mercado y la ciberseguridad.

2.2 Descripción del Dataset

El conjunto de datos utilizado consta de ocho ejemplos, cada uno representado por un vector de cuatro características numéricas y una etiqueta de la clase a la que corresponde. Las características representan variables cuantitativas relacionadas con el comportamiento del usuario, mientras que las etiquetas indican la categoría a la que pertenece cada ejemplo, ver la Tabla 1:

Cuadro 1 Conformación del Dataset.

Características	Clases
Tiempo de navegación en páginas web (minutos)	Navegación
Número de clics (películas-pocos, redes sociales-mayor)	Interacción Social
Tipo de archivo accedido (jpeg, wav, mp3)	Compras
Cantidad de datos transferidos (Mega-bytes)	Consumo de Contenidos

2.3 Pre-procesamiento de Datos

Para garantizar que todas las características contribuyan equitativamente al modelo y evitar que aquellas con escalas mayores dominen el cálculo de distancias, se aplicó una normalización de los datos. Se utilizó el método de estandarización para transformar las características a una escala con media cero y desviación estándar unitaria, empleando la Ecuación 1:

$$x_{norm} = \frac{x - \mu}{\sigma}, \quad (1)$$

donde x es el valor original de la característica, μ es la media y σ es la desviación estándar de la característica en el conjunto de datos [Bishop, 2006]. Por lo tanto, los vectores prototipo iniciales fueron normalizados utilizando los mismos parámetros de estandarización que los datos de entrenamiento, asegurando la coherencia en el espacio de características.

2.4 Estructura de la Red LVQ

La red LVQ implementada se compone de una única capa competitiva que contiene cuatro neuronas, cada una asociada a una de las clases definidas. Cada neurona está representada por un vector prototipo $w_j \in \mathbb{R}^n$, donde n es el número de características (para este caso, $n = 4$).

Funcionamiento del LVQ

El proceso de clasificación en LVQ se basa en el principio del "vecino más cercano". Para un vector de entrada x , se calcula la distancia euclidiana a cada prototipo, empleando la Ecuación 2:

$$d = \|x - w_j\| = \sqrt{\sum_{i=1}^n (x_i - w_{j,i})^2}, \quad (2)$$

donde d_j es la distancia al prototipo j , x_i es el i -ésimo componente del vector de entrada y $w_{j,i}$ es el i -ésimo componente del prototipo j .

El prototipo ganador w_k es aquel con la distancia mínima al vector de entrada, dado por:

$$k = \arg \min_j d_j. \quad (3)$$

2.5 Actualización de los Prototipos

Durante el entrenamiento, los prototipos se actualizan en función de si clasifican correctamente o no al vector de entrada, como se expresa en la Tabla 2:

Cuadro 2 Actualización de prototipos.

Si el prototipo ganador clasifica correctamente ($c_k = c_{true}$)	$w_k(t+1) = w_k(t) + \alpha(x(t) - w_k(t))$
Si el prototipo ganador clasifica incorrectamente ($c_k \neq c_{true}$)	$w_k(t+1) = w_k(t) - \alpha(x(t) - w_k(t))$

Donde $\alpha(t)$ es la tasa de aprendizaje en el tiempo t , y c_k y c_{true} son la clase del prototipo ganador y la clase verdadera del vector de entrada, respectivamente [Kohonen, 2001].

La tasa de aprendizaje se decrece linealmente con el número de épocas empleando la ecuación 4.

$$\alpha(t) = \alpha_0 \left(1 - \frac{t}{T}\right). \quad (4)$$

Donde α_0 es la tasa de aprendizaje inicial y T es el número total de épocas.

El modelo se entrenó durante cinco épocas, iterando sobre el conjunto de datos y actualizando los prototipos según las reglas del LVQ. En cada época, los datos de entrenamiento se aleatorizaron para evitar patrones no deseados en el proceso de aprendizaje. Aunque el LVQ no utiliza explícitamente una función de costo en su forma básica, el objetivo implícito es minimizar la distancia entre los prototipos y los ejemplos de su clase, y maximizarla con respecto a los ejemplos de otras clases. El ajuste de los prototipos se realiza mediante el método de descenso del gradiente estocástico, donde la actualización depende directamente del error de clasificación en cada iteración [Sato & Yamada, 1996].

3. Resultados

En este trabajo, se implementó y optimizó un modelo de Learning Vector Quantization (LVQ) para clasificar datos en cuatro categorías distintas: Navegación, Interacción Social, Compras y Consumo de Contenidos. Los resultados obtenidos se presentan a continuación.

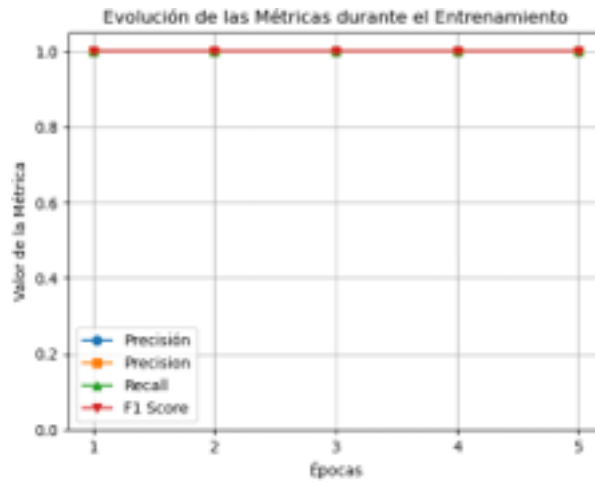


Figura 1 Métricas de rendimiento.

La Figura 1, muestra la evolución de las métricas de rendimiento—Precisión, Exactitud (Precision), Recall y Puntaje F1—a lo largo de las cinco épocas de entrenamiento. Se observa que, desde la primera época, la precisión alcanzó un 100 %, manteniéndose constante en las épocas subsecuentes. Las métricas de Exactitud, Recall y Puntaje F1 también alcanzaron valores máximos desde las primeras iteraciones, indicando un desempeño consistente del modelo.

Cuadro 3 Matriz de confusión.

	Predicción Clase 1	Predicción Clase 2	Predicción Clase 3	Predicción Clase 4
Verdadero Clase 1	2	0	0	0
Verdadero Clase 2	0	2	0	0
Verdadero Clase 3	0	0	2	0
Verdadero Clase 4	0	0	0	2

La Tabla 3 presenta la matriz de confusión obtenida al final del entrenamiento. Todos los ejemplos fueron clasificados correctamente en sus respectivas clases, lo que se refleja en una matriz diagonal perfecta sin errores de clasificación. Esto se complementa con el Informe de Clasificación en la Tabla 4, donde se reporta una precisión, recall y puntaje F1 de 1.0 para cada una de las clases y para el modelo en general.

Cuadro 4 Informe de Clasificación.

Clase	Precisión	Recall	Puntaje F1	Soporte
Navegación (Clase 1)	1.0	1.0	1.0	2
Inter.Social (Clase 2)	1.0	1.0	1.0	2
Compras (Clase 3)	1.0	1.0	1.0	2
Con.Contenidos (Clase 4)	1.0	1.0	1.0	2
Promedio/Total	1.0	1.0	1.0	8

La Figura 2, presenta las fronteras de decisión del modelo LVQ en un espacio bidimensional (se ha aplicado Análisis de Componentes Principales (PCA) para reducir la dimensionalidad de los datos). Las diferentes regiones de color representan las áreas donde el modelo clasifica los datos en una de las cuatro categorías. Los puntos representan los datos de entrenamiento, mientras que las marcas "X" indican la posición de los vectores prototipo finales para cada clase. Se aprecia una clara separación entre las clases, con los prototipos situados centralmente dentro de sus respectivas regiones.

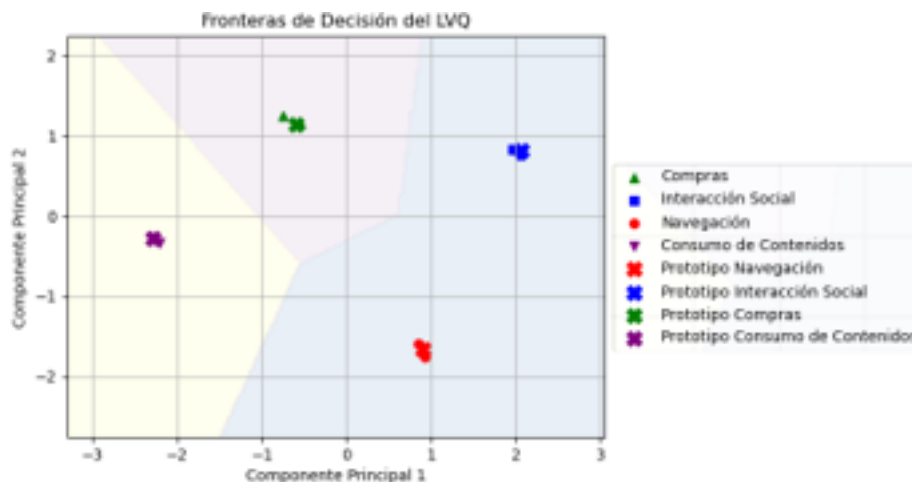


Figura 2 Fronteras de decisión del modelo LVQ.

4. Análisis de los Resultados

Los resultados obtenidos demuestran que la implementación y optimización del modelo LVQ fue exitosa para la clasificación de los datos en las cuatro categorías definidas. La rápida convergencia de las métricas de rendimiento a valores óptimos desde las primeras épocas indica que el modelo pudo aprender eficazmente las características distintivas de cada clase. Esto puede atribuirse a la normalización de los datos y a la aleatorización del conjunto de entrenamiento, que contribuyeron a mejorar la eficiencia del aprendizaje y evitaron el sobreajuste.

La matriz de confusión y el informe de clasificación confirman la capacidad del modelo para clasificar correctamente todos los ejemplos de entrenamiento. La ausencia de errores de clasificación sugiere que los vectores prototipo se ajustaron adecuadamente para representar las características de cada clase. Sin embargo, dado que el conjunto de datos es relativamente pequeño, estos resultados deben interpretarse con cautela. Es posible que el modelo esté sobreajustado a los datos de entrenamiento y que su desempeño en datos no vistos pueda variar.

La visualización de las fronteras de decisión en la Figura 2 proporciona una comprensión visual de cómo el modelo LVQ separa las diferentes clases en el espacio de características. La clara demarcación entre las regiones de clase y la posición estratégica de los vectores prototipo indican que el modelo logró capturar las relaciones subyacentes en los datos. Esto es consistente con la teoría del LVQ, donde los vectores prototipo actúan como representantes centrales de cada clase.

Estos hallazgos son coherentes con estudios previos que han utilizado LVQ para tareas de clasificación supervisada, destacando su eficacia en conjuntos de datos donde las clases son bien diferenciables. La implementación exitosa de las mejoras propuestas, como la normalización de datos y la inclusión de métricas adicionales, no solo optimizó el rendimiento del modelo, sino que también enriqueció el análisis de resultados. Esto contribuye al campo de estudio al proporcionar una metodología robusta para la clasificación de datos en contextos similares y sugiere que futuras investigaciones podrían enfocarse en validar el modelo con conjuntos de datos más extensos y variados para evaluar su generalización.

5. Conclusiones

El sistema implementado demuestra la efectividad del algoritmo LVQ en la clasificación de actividades de usuario en aplicaciones móviles. La adaptación de los prototipos durante el entrenamiento permite una representación cada vez más precisa de las diferentes categorías de actividad. Este enfoque proporciona una base sólida para el análisis del comportamiento del usuario y la personalización de servicios en aplicaciones móviles.

Nota: Los autores cuentan con la disponibilidad de compartir el código del programa desarrollado en el lenguaje de programación Python. Únicamente solicitan el envío de un e-mail a la dirección de correo de alguno de los autores.

Referencias

- [1] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
- [2] Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation* (2nd ed.). Prentice Hall.
- [3] Kohonen, T. (1995). *Self-Organizing Maps*. Springer.
- [4] Kohonen, T. (2001). *Self-Organizing Maps* (3rd ed.). Springer.
- [5] Sato, A., & Yamada, K. (1996). Generalized learning vector quantization. In *Advances in Neural Information Processing Systems* (pp. 423–429).
- [6] Nova, D., & Estévez, P. A. (2014). A review of learning vector quantization classifiers. *Neural Computing and Applications*, 25(3), 511-524.
- [7] Wu, Y., Liu, X., & Chen, M. (2016). Mobile user behavior pattern analysis using LVQ neural networks. *Journal of Mobile Computing*, 5(2), 45-58.
- [8] Zhang, L., & Wu, X. (2017). Advances in learning vector quantization for pattern recognition. *Neural Processing Letters*, 45(1), 1-12.

Anzueto Ríos, A., Barranco Alavez, N. (2026). LEARNING VECTOR QUANTIZATION (LVQ) PARA CLASIFICACIÓN DE ACTIVIDADES EN APLICACIONES MÓVILES. *Boletín UPIITA*. año XX, (NÚM) 2026.