ChatGPT

Selección de características para HMM en series temporales financieras

Para modelar regímenes de mercado con un HMM (o HSMM) sobre series OHLC, disponemos de un conjunto amplio de variables (p.ej. retornos, indicadores técnicos, estacionales, etc.). Como no tenemos etiquetas de régimen predefinidas, la selección de características debe ser **no supervisada**. En este caso no hay una "respuesta" que guíe la búsqueda; debemos evaluar la relevancia de las variables según criterios intrínsecos (p.ej. preservación de estructura, varianza, capacidad predictiva) [1]. Dado que la generación de features ya está completa, primero aplicamos filtros básicos para eliminar atributos poco útiles o redundantes [2].

Filtrado inicial de características

1. **Variables constantes o casi constantes.** Unimos/mismos valores o muy baja varianza aportan casi nada al modelo. Se puede eliminar cualquier feature con varianza cero o mayoritaria（por ejemplo, si 98 % de sus valores son iguales) [2].
2. **Duplicados exactos.** Si una variable es idéntica a otra (duplicada), una de ellas puede suprimirse sin pérdida de información [2].
3. **Características altamente correlacionadas.** Calculamos la matriz de correlación (p.ej. Pearson) entre pares de variables. Si dos features tienen correlación lineal muy alta (p.ej. $|\rho|>0.9$), aportan información redundante [3]. En tal caso, conviene descartar una de ellas: por ejemplo, si tenemos tanto "high" y "close" muy correlacionados, podremos quedarnos solo con "close" o con el retorno. En general, eliminar una variable correlacionada no reduce la capacidad predictiva del conjunto pero simplifica el modelo [3].

Estas etapas de filtrado son métodos *univariantes* o de filtro: operan sin entrenar el HMM y ayudan a recortar la dimensionalidad inicial [2]. Por ejemplo, tras esto podríamos descartar las columnas del tiempo original ("datetime") o del precio sin procesar (open/high/low) si ya usamos variables derivadas (retornos, rangos), ya que muchas de esas aportan información duplicada.

Métodos para generar subconjuntos candidatos

Con las variables restantes, podemos aplicar varios enfoques:

- **Selección por filtro multivariante.** Agrupar características (p.ej. por su correlación) y seleccionar representantes. Una técnica es la clusterización de variables: agrupar features similares y elegir una por clúster. Otra es usar análisis de componentes principales (PCA) para ver qué original features contribuyen más a la varianza. Sin embargo, PCA crea combinaciones lineales en lugar de seleccionar elementos originales, por lo que puede servir como guía pero no reemplazo de la selección de subconjunto.

- **Métodos envoltura (wrapper).** Se generan subconjuntos de prueba y se entrena un HMM con cada uno, evaluando su desempeño. Por ejemplo, un procedimiento *forward selection* podría iniciar con un feature y agregar iterativamente nuevas variables que mejoren cierta métrica. A la inversa, un *backward elimination* quitaría características de un conjunto completo. Estas técnicas

suelen ser costosas (entrenar muchos HMM) pero permiten optimizar directamente la calidad del modelo. También se pueden usar búsquedas heurísticas como algoritmos genéticos o búsqueda en árbol de decisiones para explorar subconjuntos eficientes.

- **Selección incrustada.** Algunos algoritmos permiten incorporar la selección directamente en el entrenamiento. Aunque no es común para HMM clásicos, existen variantes avanzadas donde se integra un criterio de selección en el aprendizaje. Por ejemplo, Fan y Hou (2022) proponen un modelo HMM no paramétrico con selección de características local para datos de alta dimensión [4]. En esencia, el modelo ajusta simultáneamente los parámetros del HMM y un indicador de qué variables son relevantes. Su uso concreto puede ser complejo, pero ilustra que es posible combinar modelado secuencial y selección en un solo proceso [4].

En todos los casos debemos contar con criterios objetivos para comparar subconjuntos (ver siguiente sección).

Métricas de evaluación de subconjuntos

Para decidir si un subconjunto es "bueno", usamos métricas que midan qué tanto mejora el HMM o la separación de los estados latentes:

- **Log-verosimilitud y criterios de información.** Al entrenar un HMM con cierto número de estados, la verosimilitud de los datos indicará ajuste, pero no es útil sola (siempre crece al añadir más parámetros). Por ello se emplean AIC o BIC, que penalizan la complejidad del modelo [5]. Elegimos el subconjunto de features que minimice AIC/BIC, pues esto indica mejor equilibrio entre ajuste y parsimonia. Por ejemplo, la figura siguiente muestra cómo varían AIC (azul) y BIC (verde) al cambiar el número de estados en un HMM; el mínimo señala el modelo óptimo [5].

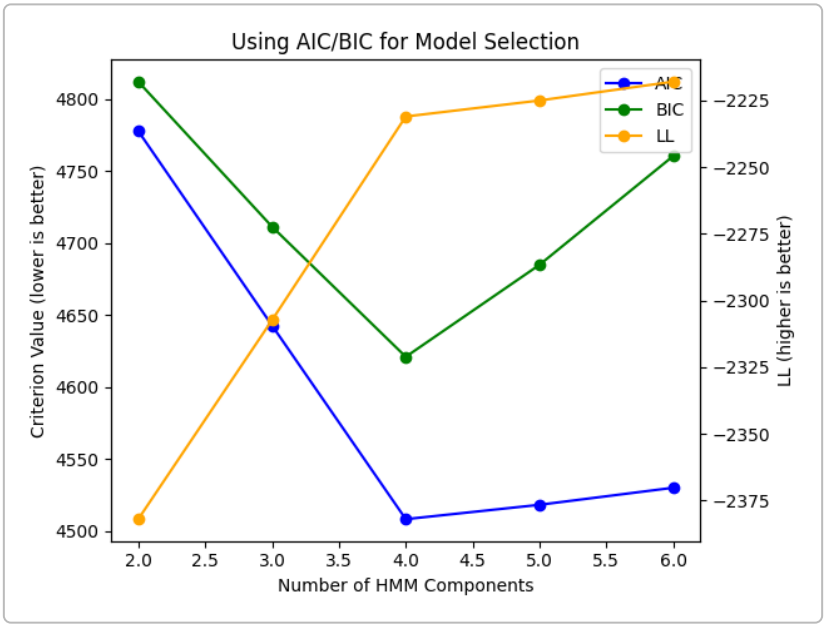

Figura: Ejemplo de criterios AIC (azul) y BIC (verde) en la selección del número de estados de un HMM (hmmlearn). El mínimo de AIC/BIC indica el modelo preferido [5].

- **Índices de agrupamiento (silhouette, Davies–Bouldin, etc.).** Dado que un HMM asigna cada instante a un estado oculto (un clúster temporal), podemos usar métricas de calidad de clústeres. El **coeficiente de silueta** mide qué tan bien separado está cada punto (instante) de los

demás en su propio estado; valores cercanos a 1 implican grupos bien definidos. El índice de **Davies–Bouldin** evalúa cuán distintos y compactos son los clústeres. Estudios recientes indican que ambos índices son muy sensibles a características irrelevantes o redundantes [6], lo que los hace útiles para comparar subconjuntos. En la siguiente imagen, por ejemplo, se ve un análisis de silueta para 2 clústeres: cada punto representa un instante, su posición horizontal muestra el coeficiente de silueta (más a la derecha = mejor agrupamiento) y los colores distinguen los clusters.

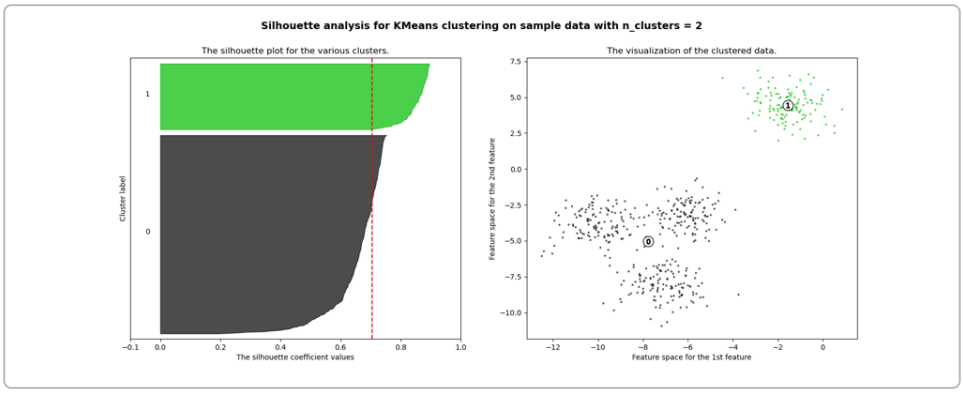

Figura: Ejemplo de gráfico de silueta para K-means (2 clusters). Las barras muestran el coeficiente de silueta de cada punto; mientras más larga hacia la derecha es una barra, mejor está separado ese punto. Índices como este ayudan a evaluar la calidad de los regímenes extraídos [6].

- **Validación cruzada temporal.** Podemos medir la verosimilitud o el error en series de prueba (hold-out) usando los estados inferidos. Dividir los datos en segmentos temporales (p.ej. años) permite validar estabilidad. Sin embargo, debe hacerse con cuidado para respetar la dependencia temporal.

- **Interpretabilidad de los estados.** Aunque no es un "número", conviene revisar si los estados resultantes tienen sentido financiero: por ejemplo, que un estado corresponda claramente a mercado alcista (bajas volatilidad, alta media móvil) y otro a bajista (alta volatilidad, rediciones negativas). Se puede comparar la media de retornos o la volatilidad dentro de cada estado. Un buen conjunto de features debería generar estados diferenciables según estas propiedades.

Al final, el criterio puede ser una combinación de los anteriores: buscamos el subset que maximiza la verosimilitud y minimiza AIC/BIC, y además produce estados bien separados (alto silueta/DB). Según McCrory y Thomas (2024), índices como la silueta son excelentes para optimizar selección no supervisada, pues caen fuertemente ante features irrelevantes [6].

Flujo de trabajo sugerido

Un posible **pipeline** de selección de variables puede ser:
- **1. Preprocesamiento**: Escalar o normalizar variables si es necesario. Convertir tiempo a variables cíclicas (ya hecho con seno/coseno). Decidir uso de precios vs. retornos (p.ej. usar solo *ret_log* en lugar de *close*, *high*, *low* por separado).
- **2. Filtrado básico**: Eliminar features constantes/quasi-constantes y duplicadas [2]. Luego aplicar filtrado de correlación: calcule la matriz de correlación y suprima uno de cada par con correlación alta [3].
- **3. Selección guiada por dominio**: Agrupar características similares (p.ej. momentum vs. volatilidad). Se puede realizar un análisis de cluster de características (por ejemplo, clustering jerárquico sobre

correlaciones) y elegir un representante de cada grupo relevante.
- **4. Búsqueda iterativa (wrapper)**: Empezar con un subset inicial (quizás las variables más prometedoras por insight o por orden de varianza explicada). Luego probar agregar o quitar features de a uno, entrenando el HMM y evaluando con AIC/BIC y medidas de clúster (silhouette/DB). Mantener cambios que mejoren las métricas. Alternativamente, emplear algoritmos genéticos o búsqueda hacia adelante/atrás para automatizar esta búsqueda.
- **5. Evaluación final**: Para el subset candidato final, validar con datos no vistos. Revisar la verosimilitud en test, AIC/BIC, y examinar la segmentación: ¿los estados capturan claramente regímenes de mercado distintos (por ejemplo, alta vs. baja volatilidad)? Se puede complementar analizando estadísticas internas de cada estado (medias, varianzas de retorno, perfil intradiario) para asegurar que tengan sentido financiero.

En resumen, la selección de features para un HMM en series temporales financieras combina pasos de filtrado estadístico y de prueba de modelo. Filtros iniciales reducen el conjunto, y luego criterios basados en el modelo (AIC/BIC, validación cruzada) junto con métricas de agrupamiento (silhouette, Davies-Bouldin) guían la elección final [3] [6]. Este proceso garantiza que el subconjunto elegido sea informativo, no redundante y produzca estados latentes coherentes con los regímenes del mercado.

Fuentes: Revisión bibliográfica sobre selección de características no supervisada [1] [4] [2] [3]; ejemplos de criterios AIC/BIC en HMM [5]; estudio de métricas de clúster para selección de features [6].

---

[1] A review of unsupervised feature selection methods | Artificial Intelligence Review
https://link.springer.com/article/10.1007/s10462-019-09682-y?error=cookies_not_supported&code=d4a97fc6-873a-44e9-a3bf-e966cf48952d

[2] [3] Hands-on with Feature Selection Techniques: Filter Methods - Fritz ai
https://fritz.ai/hands-on-with-feature-selection-techniques-filter-methods/

[4] Unsupervised modeling and feature selection of sequential spherical data through nonparametric hidden Markov models | OpenReview
https://openreview.net/forum?id=A7AmSkGvaj

[5] Using AIC and BIC for Model Selection — hmmlearn 0.3.3.post1+ge01a10e documentation
https://hmmlearn.readthedocs.io/en/latest/auto_examples/plot_gaussian_model_selection.html

[6] [2402.12008] Cluster Metric Sensitivity to Irrelevant Features
https://arxiv.org/abs/2402.12008