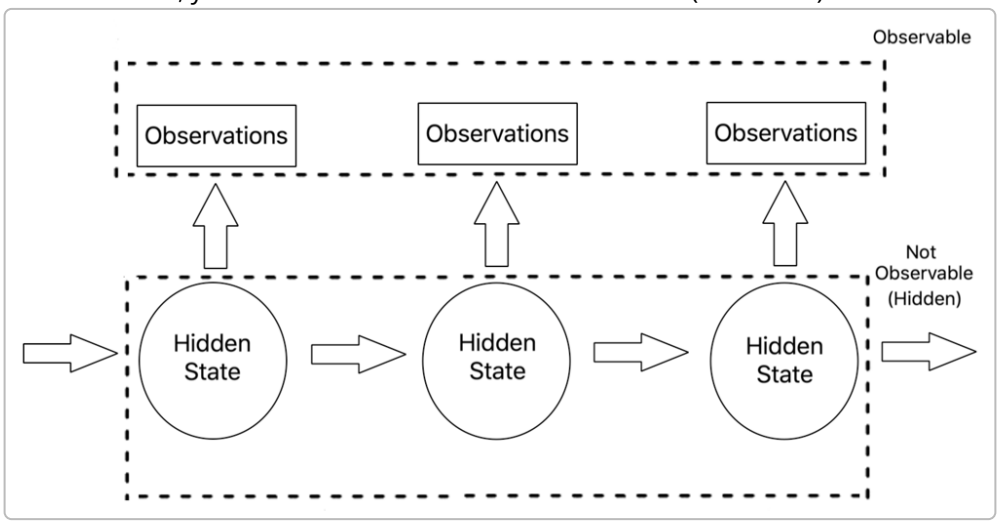
ChatGPT

Introducción: Los modelos de **cambio de régimen** son herramientas esenciales para capturar dinámicas financieras no estacionarias, donde las series temporales (por ejemplo, retornos diarios, volatilidad, etc.) alternan entre distintos comportamientos estadísticos ("regímenes"). En este informe se exploran **técnicas avanzadas** para modelar cambios de régimen en datos OHLCV (Open-High-Low-Close-Volume) usando **Modelos de Markov Ocultos (HMM)** y **Modelos de Estado Conmutables (SSSM)**. Se cubren los fundamentos matemáticos de estos modelos, extensiones con distribuciones heavy-tail (*Student-t*), métodos de inferencia variacional para escalabilidad, técnicas para determinar el número óptimo de regímenes (incluyendo enfoques no paramétricos Bayesianos), estrategias para incorporar información macroeconómica de forma suave, y un pipeline completo de aplicación a datos financieros con visualizaciones, pseudocódigo y referencias a la literatura.

1. Fundamentos matemáticos de HMM y SSSM (con emisiones Student-t)

Un **Modelo Oculto de Markov (HMM)** describe un proceso donde existe una secuencia de **estados ocultos** z_t (no observables directamente) que siguen una cadena de Markov de primer orden, y por cada estado oculto se genera una **observación** visible y_t en el tiempo t [1] [2]. En forma simple, un HMM se define por: (a) un conjunto finito de estados ocultos $z_t \in {1,\dots, K}$, (b) una matriz de probabilidades de transición $A = [a_{ij}]$ con $a_{ij}=P(z_{t}=j \mid z_{t-1}=i)$, (c) una distribución de **emisión** $P(y_t \mid z_t=j)$ para cada estado j, y (d) una distribución inicial $P(z_0)$ [3] [4]. La **asunción de Markov** implica que $P(z_t \mid z_{t-1}, z_{t-2},\dots) = P(z_t \mid z_{t-1})$, y además las observaciones son condicionalmente independientes dado el estado actual, $P(y_t \mid z_t, z_{1:t-1},y_{1:t-1}) = P(y_t \mid z_t)$ [2]. La Figura 1 ilustra la arquitectura de un HMM: los estados ocultos (círculos) evolucionan con transiciones Markovianas, y cada estado emite una observación visible (cuadrados)



. Matemáticamente, la probabilidad conjunta de estados y observaciones para una secuencia de longitud T es:

$$P(z_{0:T},\,y_{0:T}) \;=\; P(z_0)\prod_{t=1}^{T}P(z_t \mid z_{t-1})\,P(y_t \mid z_t)\,. $$

Esta formulación captura regímenes ocultos que "generan" las observaciones financieras, p. ej. un estado oculto podría corresponder a un régimen de baja volatilidad que emite retornos pequeños, mientras otro estado representa un régimen de alta volatilidad con retornos extremos.

En el **HMM Gaussiano clásico**, las emisiones y_t suelen modelarse con distribuciones Gaussianas condicionadas al estado (ej.: $y_t \mid z_t=j \sim \mathcal{N}(\mu_j,\Sigma_j)$). Sin embargo, para datos financieros es habitual observar colas pesadas (eventos extremos más frecuentes que lo que predice la Normal). Por ello, sustituir las Gaussianas por **distribuciones *Student-t*** mejora la robustez ante outliers y riesgos extremos [5] [6]. Un HMM con emisiones *Student-t* (denotado a veces *Student's t-HMM* o *SHMM*) asume $y_t \mid z_t=j$ sigue una distribución *t* de Student con ciertos grados de libertad ν_j, media μ_j y escala Σ_j. La densidad para una variable d-dimensional sería, por ejemplo:

$$ P(y_t \mid z_t=j) = \frac{\Gamma((\nu_j+d)/2)}{\Gamma(\nu_j/2)\,(\nu_j\pi)^{d/2}|\Sigma_j|^{1/2}} \Big(1 + \frac{1}{\nu_j}(y_t-\mu_j)^T\Sigma_j^{-1}(y_t-\mu_j)\Big)^{-(\nu_j+d)/2}.$$

Esta distribución tiene colas más pesadas controladas por ν_j: a menor ν, más pesada la cola (si $\nu \to \infty$, la *t* tiende a la Normal). En la práctica, implementar emisiones *t* en HMM se facilita mediante la representación de *Student-t* como una **mezcla de escala-gaussiana** (escala-mixture): una variable *t* se puede expresar como $y = \mu + \sigma \sqrt{w}\,u$ donde $u \sim \mathcal{N}(0,I)$ y w es una variable latente con distribución Inversa-Gamma [7] [5]. Este truco introduce un parámetro latente de escala w_t para cada observación, convirtiendo el modelo en un "HMM Gaussiano-Gamma" donde, dado w_t, $y_t|z_t=j,w_t \sim \mathcal{N}(\mu_j,\,\Sigma_j/w_t)$ y $w_t|z_t=j \sim \text{Gamma}(\nu_j/2,\nu_j/2)$ [8] [9]. Así se obtiene una forma cerrada para las ecuaciones de inferencia manteniendo la exponencial-conjugación, permitiendo aplicar algoritmos similares a los del caso Gaussiano pero capturando heavy-tails.

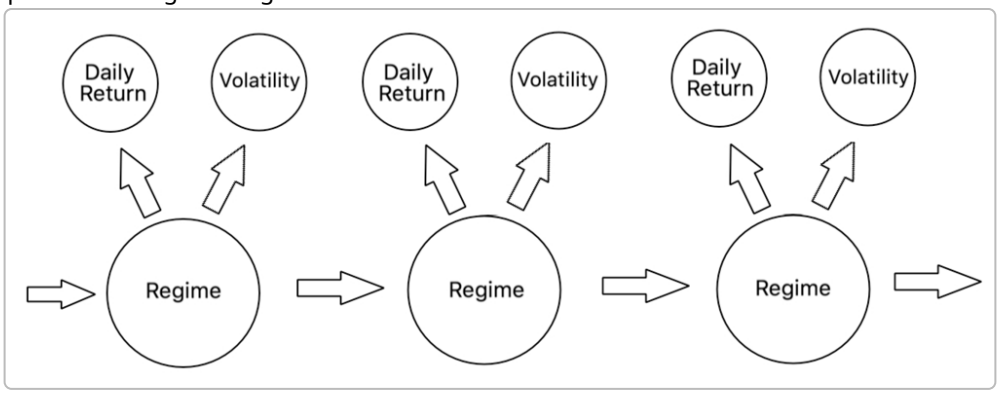
Ventajas de las emisiones *Student-t*: Estudios han demostrado que los HMM con distribuciones *t* pueden identificar regímenes financieros más persistentes y realistas en presencia de datos con outliers o volatilidad abrupta [10]. En particular, Bulla (2011) encuentra que un HMM-t tiende a producir estados ocultos más estables en series temporales financieras que un HMM Gaussiano, ya que la cola pesada reduce la necesidad de crear cambios de estado para explicar observaciones extremas [10]. Esto resulta muy apropiado para mercados donde eventos como "crashes" o "rallies" son más frecuentes de lo normal.

Por su parte, un **Modelo de Estado Espacio Conmutable (SSSM)** (también conocido como **"Switching State-Space Model"** o **"Switching Linear Dynamical System"**) extiende los HMM introduciendo dinámicas continuas dentro de cada régimen [11]. En un SSSM coexisten dos niveles de estados ocultos: uno **discreto** z_t que representa el régimen (como en el HMM), y otro **continuo** x_t que sigue un modelo de espacio de estado lineal (p. ej. un filtro de Kalman) cuyas matrices dependen del régimen z_t. En otras palabras, cuando $z_t = j$, el sistema dinámico (por ejemplo, un proceso AR(1) multivariante, o un modelo de factor latente) evoluciona con parámetros específicos del régimen j. Un ejemplo típico es un **modelo de Markov conmutado de medias y varianzas**:

- *Ecuación de estado (dinámica latente):* $x_t = A_{z_t}\,x_{t-1} + B_{z_t}\,u_t$, donde u_t es ruido Gaussiano (o *Student-t* latente para heavy-tails) y A_{z_t}, B_{z_t} matrices que pueden cambiar con z_t.
- *Ecuación de observación:* $y_t = C_{z_t}\,x_t + D_{z_t}\,v_t$, con v_t ruido de observación.

Así, cada régimen $z_t=j$ define un submodelo lineal diferente (por ejemplo, un régimen podría ser un **mercado alcista** donde x_t sigue una tendencia con cierta volatilidad, y otro un **mercado bajista** donde x_t sigue una dinámica distinta). Al cambiar z_t, el modelo "conmuta" entre distintos sistemas dinámicos [12] [13]. En el caso especial donde x_t no tiene dinámica (o es trivial) y las observaciones dependen directamente de z_t, el SSSM se reduce al HMM estándar. Por eso se dice que los modelos de estado conmutable **generalizan** a los HMM (añadiendo estado continuo) y a los modelos lineales dinámicos (añadiendo regímenes discretos) [13]. Este enfoque es muy utilizado en econometría – p. ej., el famoso modelo de **cambio de régimen de Hamilton (1989)** es un caso particular, donde se alternan ecuaciones AR según un estado de Markov oculto para modelar expansiones y recesiones económicas.

Un ejemplo ilustrativo de SSSM es el **Filtro de Kalman conmutable**, en el cual un HMM selecciona entre M posibles modelos lineales de Kalman. La Figura 2 muestra esquemáticamente cómo un HMM de 3 estados (regímenes) podría emitir dos observables – digamos *retorno diario* y *volatilidad estimada* – cuyos valores dependen del régimen vigente



. Nótese que en cada régimen (círculo "Regime") las observaciones (p. ej. retorno y volatilidad) se distribuyen de forma distinta. Este tipo de modelado permite captar, por ejemplo, que en el **Régimen 1** los retornos tengan media positiva y baja varianza (bull market tranquilo), en el **Régimen 2** retornos con media cercana a cero y volatilidad baja (mercado lateral), y en el **Régimen 3** retornos con media negativa y volatilidad muy alta (crash o panic market).

Formalismo matemático del SSSM: Una forma compacta de describir un SSSM es mediante la probabilidad conjunta:

$$P(z_{0:T},x_{0:T},y_{0:T}) \;=\; P(z_0)\,P(x_0 \mid z_0)\,\prod_{t=1}^{T} P(z_t \mid z_{t-1})\; P(x_t \mid x_{t-1}, \,z_t)\; P(y_t \mid x_t,\,z_t)\,. $$

Aquí $P(x_t \mid x_{t-1}, z_t=j)$ define la dinámica lineal en el régimen j, y $P(y_t \mid x_t, z_t=j)$ la distribución de observación en ese régimen. Si las distribuciones son Gaussianas (o *Student-t*), y las funciones lineales, la inferencia exacta se vuelve intratable debido a la combinación discreto-contínua (el número de posibles trayectorias crece exponencialmente con T) [14] [15]. Por ello, se recurre a técnicas aproximadas (ver Sección 2). A pesar de la complejidad, los SSSM son muy poderosos: combinan la capacidad de detectar **cambios cualitativos** (régimen alto/bajo riesgo) con la capacidad de modelar **dinámicas cuantitativas locales** (tendencias, autocorrelaciones de corto plazo, etc.).

2. Inferencia variacional para HMM/SSSM: escalabilidad y eficiencia

El aprendizaje de un HMM tradicionalmente se realiza con el algoritmo de **Baum-Welch**, que es un caso particular del **EM (Expectation-Maximization)** para modelos con variables latentes de tipo cadena de Markov [16]. En EM, iteramos: (E) calcular esperanzas a posteriori de estados ocultos dado los parámetros actuales (usando *forward-backward*), y (M) re-estimar los parámetros que maximizan la verosimilitud esperada. EM garantiza convergencia a un óptimo local de la verosimilitud. Sin embargo, **EM presenta limitaciones**: (i) puede atascarse en óptimos locales pobres si la inicialización es desfavorable, (ii) es propenso a sobreajuste en modelos con muchos parámetros (especialmente si no hay regularización), y (iii) en algunos modelos complejos (como SSSM con muchas dimensiones latentes) la E-step exacta es intractable. Además, la complejidad de EM crece con $O(K^2 T)$ para un HMM de K estados y secuencia de longitud T, lo cual puede ser costoso para K grande o *T* muy grande.

Inferencia variacional Bayesiana ofrece una alternativa escalable y más robusta. En lugar de estimar parámetros puntuales por máxima verosimilitud, adoptamos un enfoque Bayesiano donde asignamos distribuciones a los parámetros y estados ocultos, y luego buscamos una aproximación q para la distribución posterior $p(z_{0:T},\Theta \mid \text{datos})$ (donde Θ denota parámetros) [17] [18]. La idea central es maximizar la *evidencia variacional* $ \mathcal{L}(q) = E_q[\log p(y,z,\Theta)] - E_q[\log q(z, \Theta)]$, que es un lower bound de $\log p(y)$, mediante restricción de q a una familia tractable (ej. *mean-field*, donde asumimos factorizar q sobre distintas variables). En HMM, una elección típica es factorizar q como $q(\Theta)\,q(z_{0:T})$ o incluso $q(\Theta)\prod_t q(z_t)$ con ciertas dependencias temporales (en cadenas, no se suele factorizar completamente z_t independientes, sino emplear aproximaciones estructuradas respetando la cadena de Markov). El algoritmo resultante, llamado a veces **Variational Bayes EM** o **Bayesiano variacional**, realiza pasos similares a EM pero actualizando distribuciones en lugar de valores puntuales:

- **Paso E Variacional:** Calcular $q(z_{0:T})$ aproximadamente. En un HMM, esto se hace pasando *mensajes* hacia adelante y atrás como en forward-backward, pero usando los parámetros actuales y *priors* en lugar de conteos absolutos [19] [20]. Se obtienen distribuciones posteriores aproximadas $ \gamma_t(i) = q(z_t=i)$ (responsabilidades) y $\xi_t(i,j)=q(z_{t-1}=i,z_t=j)$ (contabilizando transiciones). *Nota:* En inferencia variacional de HMM, aplicar forward-backward es válido bajo ciertas condiciones de normalización de los parámetros; trabajos recientes han demostrado rigurosamente su corrección en el contexto variacional [21] [22].
- **Paso M Variacional:** En lugar de MLE de parámetros, actualizamos $q(\Theta)$, típicamente perteneciente a distribuciones conjugadas. Por ejemplo, para probabilidades de transición se suele asumir un prior de Dirichlet α; tras la E-step, la distribución posterior de cada fila i de la matriz de transición viene dada por $q(a_{i,\cdot}) = \text{Dirichlet}(\alpha + N_{i,\cdot})$ donde $N_{i,j}= \sum_{t}\xi_t(i,j)$ es el conteo esperado de transiciones $i\to j$. Análogamente, si modelamos las emisiones con distribuciones con parámetros θ_j y prior $p(\theta_j)$ conjugado, su posterior $q(\theta_j)$ se actualiza usando las suficientes estadísticas acumuladas ponderadas por $\gamma_t(j)$. Estos pasos se repiten hasta convergencia del bound $\mathcal{L}(q)$.

La inferencia variacional provee varias ventajas: **(a)** Es **más estable numéricamente** que EM (evita singularidades, p. ej. varianzas que colapsan a 0, mediante priors no informativos) [23] [24]. **(b)** Introduce regularización bayesiana, mitigando sobreajuste; por ejemplo, los parámetros de emisión y transición no divergen, sino que permanecen regidos por sus distribuciones posteriores que integran la incertidumbre.

(c) Facilita la **extensión a modelos complejos**: en SSSM donde EM exacto es intratable, el variacional permite aproximar un filtro conjunto de z_t y x_t. Ghahramani y Hinton (2000) desarrollaron un algoritmo variacional para SSSM que combina recursiones de Kalman y HMM, logrando inferencia tractable en el modelo híbrido [25] [26]. En esencia, aproximan la posterior conjunta $q(z_{0:T},x_{0:T})$ factoriza en $q(z_{0:T})\,q(x_{0:T})$ con ciertas dependencias, realizando una iteración donde se calculan esperanzas de x_t con un filtro/smoother Kalman ponderado por la distribución de z (esto es, *smooth Kalman filtering* condicionado a un peso de pertenencia a cada régimen) [25]. Si bien los detalles matemáticos son extensos, el resultado práctico es que la inferencia variacional hace posible el entrenamiento de modelos con decenas de estados y múltiples variables en un tiempo razonable, donde EM clásico fracasaría o sería muy lento.

Un aporte crucial de la inferencia Bayesiana variacional es la **capacidad de determinar el número de regímenes automáticamente** (ver sección 3). Chatzis et al. (2010) propusieron un esquema variacional para HMM con emisiones *Student-t* que ilustra estos beneficios: su método *VB-HMM-t* fue más robusto que EM a outliers y convergió a modelos de tamaño óptimo sin necesidad de validación cruzada [17] [24]. En resumen, la inferencia variacional convierte el entrenamiento de HMM/SSSM en un problema de optimización determinista sobre distribuciones, evitando las simulaciones costosas de MCMC pero capturando parte de la incertidumbre de un enfoque plenamente Bayesiano.

3. Detección automática del número óptimo de regímenes

¿Cuántos regímenes tiene el mercado? Elegir el número de estados ocultos K es un desafío clave en modelado de regímenes. Existen enfoques **frecuentistas** basados en criterios de información y enfoques **bayesianos no paramétricos** que dejan que los datos "decidan" K.

(a) Criterios de información (AIC, BIC, VB-BIC): Una estrategia clásica es entrenar varios HMM con diferente K y escoger aquel con mejor criterio de penalización, como **BIC (Bayesian Information Criterion)**. El BIC para un modelo dado es $\text{BIC} = -2\log L_{\max} + p \log N$, donde $L_{\max}$ es la máxima verosimilitud del modelo, p el número de parámetros estimados y N el número de observaciones. En contextos HMM, $N=T$ (longitud de la serie) y p incluye las transiciones ($K(K-1)$ libres si ninguna prohibida) más parámetros de emisión (por ejemplo K medias y varianzas univariadas, etc.). Un BIC más bajo indica mejor balance ajuste/complejidad. También puede usarse el **AIC** ($-2\log L + 2p$). Ambos criterios tienden a penalizar fuertemente modelos con muchos estados que solo marginalmente mejoran la verosimilitud, lo que ayuda a evitar sobreajuste. En la práctica, para datos financieros, suele observarse que el BIC selecciona un número de regímenes relativamente pequeño (2–4) a menos que haya evidencia contundente de más regímenes. Sin embargo, estos criterios asumen *MLE* tradicional. En marco variacional, se puede derivar un análogo de BIC usando la *evidencia variacional*. De hecho, la función objetivo variacional $\mathcal{L}(q)$ es un lower bound de $\log p(\text{datos}|\text{modelo})$. Puede compararse $\mathcal{L}$ penalizando la complejidad modelo en forma similar a BIC (a veces llamado **VB-BIC**): $-2\mathcal{L}_{\max} + p \log N$. Una propiedad interesante del enfoque variacional es que tiende a "apagar" estados innecesarios: por ejemplo, los pesos de Dirichlet para transiciones de ciertos estados pueden colapsar a valores bajos, implicando que esos estados prácticamente no se utilizan (posterior de z_t casi nunca los elige). En consecuencia, **la inferencia variacional a menudo realiza selección de modelo automática**; Chatzis (2011) reporta que su HMM variacional con emisiones *t* determinó el tamaño óptimo de forma interna, sin cross-validation [23] [24]. No obstante, en aplicaciones críticas conviene corroborar con métodos adicionales (p. ej., comparar log-verosimilitudes de predicción out-of-sample).

(b) Modelos no paramétricos (HDP-HMM): Enfoques bayesianos no paramétricos permiten tratar K efectivamente como infinito, dejando que la inferencia encuentre cuántos estados se *necesitan* dados los datos. El método más conocido es el **HDP-HMM (Hierarchical Dirichlet Process Hidden Markov Model)** de Teh et al. (2006) [27]. La idea es asignar un proceso de Dirichlet (DP) como prior sobre la distribución de transición. Un DP es esencialmente una distribución sobre distribuciones discretas con soporte potencialmente infinito. En un HDP-HMM, cada fila i de la matriz de transición A tiene un prior DP *condicionado* por una medida de base común G_0 que a su vez sigue un DP. Intuitivamente, esto crea un **conjunto (a priori infinito) de posibles estados**; los datos observarán solo un número finito de ellos, pero no se fija un K máximo. Formalmente, $ \beta \sim \text{GEM}(\gamma)$ (stick-breaking) genera pesos infinitos, y para cada estado i se genera $\pi_i \sim \text{DP}(\alpha,\beta)$, donde π_i son las probabilidades de transición desde i [28] [29]. Este marco se conoce también como **iHMM (Infinite HMM)**. La inferencia típicamente se hace vía MCMC (samplers tipo Gibbs bloqueado, véase Fox et al. 2011). Un inconveniente de HDP-HMM básico es que, sin restricciones, puede sobre-segmentar secuencias rápidas: tender a crear estados "redundantes" que capturan pequeñas variaciones y alternar muy frecuentemente [30] [31]. Para contrarrestar eso, Fox et al. (2010) introdujeron el **sticky HDP-HMM**, añadiendo un hiperparámetro que incrementa la probabilidad *a priori* de auto-transición (a_{ii}), fomentando duraciones más largas en cada estado [32] [33]. Este modelo ha sido exitoso en problemas de **diarización de locutor** y en finanzas para detectar **régimenes persistentes** (p. ej., separar mercados bull/bear duraderos sin sobreestimar regímenes temporales pasajeros). En resumen, el enfoque HDP-HMM ofrece una solución elegante: no necesitamos especificar K, solo definimos priors adecuados y el resultado es una distribución posterior sobre el número de estados distintos utilizados. En la práctica, el MCMC propone nuevos estados si los datos sugieren patrones no explicables con los existentes, o elimina/fusiona estados que no aportan. Alternativamente, existen implementaciones variacionales del iHMM [34] [35], que combinan la ventaja de selección automática con la eficiencia de VB.

(c) Validación y criterios ad hoc: Además de los anteriores, los practicantes suelen validar visualmente y con métricas de ajuste. Por ejemplo, **log-verosimilitud predictiva**: dividir datos en training/test, entrenar HMM con varios K en training, y evaluar prob. de los datos de test; el K que maximice esta probabilidad predictiva es preferible (similar a validación cruzada en series temporales). Otras métricas: comparar distribuciones empíricas de observables bajo segmentaciones distintas, o evaluar si los regímenes estimados tienen interpretación económica (p. ej., comprobar si un régimen corresponde mayormente a períodos de recesión conocidas). En la Figura 3 (tomada de Wang et al. 2020) se ilustra la validación de número de regímenes: los autores eligen $K=3$ justificando que más de 3 no mejora sustancialmente las métricas [36] [37]. En general, un buen protocolo es combinar enfoques: usar BIC/AIC/VB para sugerir un rango, probar HDP-HMM si es factible computacionalmente, y asegurarse de que el resultado tenga sentido financiero.

4. Incorporación de condicionamiento suave a etiquetas macroeconómicas

Los modelos discutidos hasta ahora se consideran **no supervisados**: descubren regímenes basándose solo en patrones estadísticos de la serie financiera, sin guía externa. Sin embargo, en análisis financiero es común disponer de **información macroeconómica o cualitativa** que describe el entorno de mercado (p. ej., fases de ciclo económico: expansión vs recesión, políticas monetarias, indicadores de estrés financiero, etc.). Incluir esta información **podría mejorar la detección de regímenes**, alineándolos con condiciones económicas reales, pero debe hacerse cuidadosamente para no destruir la naturaleza de descubrimiento

automático del modelo. La idea de *condicionamiento suave* se refiere a incorporar las etiquetas macroeconómicas de forma no determinante – es decir, influir en los regímenes inferidos sin forzarlos rígidamente a las etiquetas.

Existen varias aproximaciones para lograr esto:

(a) HMM no-homogéneo con covariables: Una técnica bien estudiada es permitir que las **probabilidades de transición** (e incluso las de emisión) dependan de **variables externas** (covariables). En un **HMM no-homogéneo** (NHMM), en lugar de a_{ij} fijo, modelamos $P(z_t=j \mid z_{t-1}=i,\,x_t)$ como función de un vector de covariables x_t (que podría incluir indicadores macro en tiempo t). Por ejemplo, para transiciones se puede usar una regresión logística multinomial:

$$P(z_t=j \mid z_{t-1}=i,\;x_t) \;=\; \frac{\exp(\beta_{ij}^\top x_t)}{\sum_{k=1}^K \exp(\beta_{ik}^\top x_t)}\,,$$

donde β_{ij} son coeficientes a estimar [38] [39]. De forma análoga, los parámetros de emisión (p. ej., la media de retornos en el régimen j) podrían depender de variables macro ($y_t \mid z_t=j, x_t$ con media = $\mu_j + \gamma_j^\top x_t$ por decir algo). Este enfoque, implementado en un marco Bayesianamente, permite incluso hacer **selección de variables**: determinar qué covariables macro influyen significativamente en qué transiciones/emisiones [40] [38]. Wang et al. (2023) usan un HMM no-homogéneo para series de conteos de epilepsia: incorporan decenas de covariables clínicas en las transiciones mediante funciones logit, y en las emisiones (distribución binomial negativa), aplicando además priors de selección para identificar cuáles covariables importan [41] [42]. En contexto macro-financiero, uno podría incluir variables como tasa de interés, inflación, curva de rendimientos, índice de confianza, etc., para influir en la probabilidad de cambiar de régimen. *Ejemplo:* Si la covariable "recesión (0/1)" está activa, el modelo podría aprender que la probabilidad de transitar al régimen "bear market" aumenta. Importante: esto sigue siendo "suave" porque el modelo *puede* ignorar la covariable si los datos no la respaldan suficientemente (los coeficientes β podrían salir aproximadamente cero).

(b) Priors jerárquicos e información parcial: Otra manera de incorporar etiquetas macro es estructurar el modelo en **dos niveles**: un nivel macro que evoluciona según datos conocidos, y un nivel micro (regímenes financieros) condicionado por el macro. Por ejemplo, supongamos que tenemos una etiqueta binaria macro $M_t \in {\text{Expansión, Recesión}}$ proveniente del NBER u otra fuente. Podemos construir un modelo **jerárquico** donde M_t (observado) influye con un prior sobre z_t: $P(z_t \mid z_{t-1}, M_t)$. En términos de implementación, si sabemos por expertos que ciertos regímenes ocultos probablemente correspondan a recesiones, podemos **sesgar las transiciones**: por ejemplo, hacer que si $M_t=\text{Recesión}$, la probabilidad *a priori* de estar en un estado de alta volatilidad sea mayor. Una forma bayesiana de lograrlo es mediante un prior a priori en $z_{1:T}$ que penalice discordancias con $M_{1:T}$ (p.ej., una "energía" que suma penalizaciones cada vez que z_t no coincide con la etiqueta macro esperada). Esto se puede integrar en inferencia variacional o MCMC mediante técnicas de **posterior regularization**, que imponen restricciones suaves en la distribución posterior. Así, el modelo sigue determinado primariamente por los retornos observados, pero las etiquetas macro empujan sutilmente la solución hacia una congruente con ellas.

Otra variante es proporcionar algunas **etiquetas parciales**: si ciertas fechas sabemos (o asumimos) el régimen macro, podemos anclar el estado oculto en esas fechas. Por ejemplo, si *Mayo 2020* se etiqueta como "crisis COVID", podríamos inicializar o fijar $z_{\text{mayo 2020}}$ en un estado específico asociado a crisis. Luego, el entrenamiento del HMM aprenderá a extender ese estado a otros periodos similares. Esto

convierte el aprendizaje en **semi-supervisado** (algunos z_t "etiquetados" con confianza, el resto no). Cabe destacar que las etiquetas macroeconómicas suelen ser *imperfectas* para definir regímenes financieros – de ahí que no se quiera supervisar 100%. Por ejemplo, puede haber recesiones suaves donde el mercado no cae en un régimen bajista severo, o eventos de mercado no relacionados con recesión (flash crash, etc.). Por lo tanto, incorporar macro con suavidad ayuda a **combinar conocimiento experto con descubrimiento de datos**.

(c) Ejemplo práctico: Supongamos que queremos que nuestro modelo de regímenes tenga en cuenta la política monetaria. Podemos incluir como covariable la desviación del índice de condiciones financieras de la Fed, o una variable dummy de *QE on/off*. En un HMM no-homogéneo, esto podría resultar en coeficientes β_{ij} que indiquen, por ejemplo, que durante periodos de liquidez abundante (QE on) la probabilidad de permanecer en régimen bull (i->i) aumenta, mientras que la transición a un régimen defensivo disminuye. Al mismo tiempo, las emisiones (retornos) podrían depender de la covariable: quizá en QE los retornos medios son mayores incluso dentro del mismo régimen de volatilidad. Todo esto se calibraría automáticamente en el entrenamiento. Wang et al. (2023) demuestran la eficacia de este tipo de modelo mostrando cómo ciertas covariables clínicas alteran significativamente las probabilidades de cambio de estado en pacientes [43] [44]. En finanzas, esto equivaldría a identificar qué variables macro "manejan" cambios de régimen (p. ej., la *yield curve inversion* anticipando un salto a régimen de crisis).

Conclusión de esta sección: Incorporar etiquetas macroeconómicas de forma suave es factible mediante extensiones de HMM/SSSM que incluyan covariables en transiciones/emisiones o estructuras jerárquicas. Es esencial calibrar la influencia: demasiada confianza en las etiquetas puede forzar el modelo y hacerle perder capacidad predictiva (sobre todo si las etiquetas son ruidosas o tardías, como las fechas oficiales de recesión), mientras que ignorarlas desperdicia conocimiento útil. Un enfoque bayesiano bien diseñado (por ejemplo, con priors que reflejen nuestra creencia en las etiquetas con cierta varianza) puede lograr el equilibrio. Esto alinea los regímenes ocultos con interpretaciones económicas, facilitando su uso en decisiones de inversión o gestión de riesgo alineadas con el panorama macro.

5. *Pipeline* completo de aplicación a datos OHLCV (sin *order book*)

A continuación, se describe un **pipeline de extremo a extremo** para aplicar estos modelos a datos financieros de barras OHLCV (precio de apertura, máximo, mínimo, cierre y volumen), asumiendo que no se cuenta con información de libro de órdenes. El objetivo es detectar cambios de régimen y generar **señales estructurales** útiles (por ejemplo, para ajustar una estrategia de trading o de gestión de portafolio según el régimen de mercado). El pipeline abarca: *ingeniería de características*, *modelado*, *inferencia*, *detección de cambios de régimen*, *generación de señales* y *evaluación*.

5.1 Ingeniería de características (feature extraction)

El primer paso es transformar los datos OHLCV en **características cuantitativas** que alimentarán el modelo. Dado que no tenemos datos de microestructura, nos enfocamos en características a nivel de barra diaria (o intradiaria si aplica). Ejemplos comunes de *features* útiles para detección de regímenes:

- **Retornos logarítmicos:** $r_t = \log(\text{Close}*t/\text{Close})$. Capturan cambios relativos de precio, que son la base para distinguir regímenes de rendimiento alto vs bajo.
- **Volatilidad realizada o implícita:** Se puede estimar la volatilidad diaria mediante, por ejemplo, el rango intra-día. Una fórmula clásica es la volatilidad de Parkinson: $\sigma_{\text{Parkinson},t}^2 =

\frac{1}{4\ln 2}(\ln(\text{High}_t/\text{Low}_t))^2$. Alternativamente, usar un ATR (Average True Range) normalizado por precio, o la desviación estándar de retornos intradía si se dispone.

- **Volumen (o liquidez):** Volumen negociado v_t (quizá en log) o ratio volumen/volumen medio. Regímenes de pánico suelen ir acompañados de volúmenes anómalos.
- **Indicadores técnicos suaves:** Aunque no son fundamentales, algunos pipelines incluyen indicadores para capturar momentum o reversión de media, e.g. pendiente de una media móvil, RSI, etc. Sin embargo, estos pueden ser redundantes si el modelo de estado ya capta autocorrelaciones; se deben manejar con cuidado para no mezclar señales de trading dentro del modelo de régimen.
- **Características derivadas de OHLC:** Por ejemplo, relación cierre-apertura ($\text{Close}_t - \text{Open}_t$) para medir sesgo intradía, o *candlestick features* (como patrones de vela, tamaño de mecha vs cuerpo) para identificar días de alta volatilidad intradía.

Una vez definidas las features, se construye la matriz de datos $X_{T \times d}$ donde d es el número de características seleccionadas. Normalizar o estandarizar es recomendable: típicamente los retornos se centran en 0, la volatilidad en escala log, etc., para que variables con distintas unidades sean comparables. A modo de pseudocódigo, la extracción de features podría verse así:

```
# Supongamos `data` es un DataFrame con columnas:
'Open','High','Low','Close','Volume'
features = []
prev_close = None
for index, row in data.iterrows():
    if prev_close is None:
        prev_close = row['Close']
        continue
    ret = math.log(row['Close']/prev_close)
    range_parkinson = (math.log(row['High']/row['Low']))**2
    volatility = range_parkinson / (4 * math.log(2))
    vol_norm = math.log(1 + row['Volume'])  # log-volumen
    features.append([ret, volatility, vol_norm])
    prev_close = row['Close']
X = np.array(features)
```

En este ejemplo, X_t incluiría retorno, volatilidad realizada y log-volumen para cada día.

5.2 Definición del modelo (HMM/SSSM con emisiones Student-t)

Con las features listas, definimos nuestro modelo de régimen. Debemos elegir entre HMM (más simple) o SSSM (más complejo, para capturar dinámicas internas). A falta de datos de libro de órdenes, generalmente un HMM puede ser suficiente para detectar regímenes de volatilidad/rendimiento, pero si quisiéramos modelar también autocorrelaciones en la serie de retornos podríamos usar un SSSM (p. ej., un **Markov Switching AR(1)**). Aquí optaremos por un HMM de K estados ocultos, con emisiones vectoriales $(r_t, \sigma_t,\log v_t)$ siguiendo una distribución *Student-t* multivariante.

Parámetros a configurar:

- Número de estados K: Podemos comenzar con un K estimado (por ejemplo 3: baja vol + alcista, alta vol + bajista, media vol neutral) y luego ajustar o usar HDP-HMM para decidir.
- Distribución de emisiones: Student-t multivariada. Para simplificar, podríamos asumir independencia entre features en cada estado y usar *Student-t* univariadas por feature (aunque idealmente se modela la correlación entre r_t y σ_t). Alternativamente, una mezcla de Gaussianas puede usarse si se sospecha multimodalidad, pero usualmente cada régimen se asocia con una única "clase" de comportamiento aproximadamente unimodal en retornos.
- Priors bayesianos: Elegimos priors no informativos o débiles. Ej: prior de Dirichlet($1,\dots,1$) para filas de transición (lo que equivale a asumir a priori que todas las transiciones son igual probables), e *priors* de Inversa-Gamma para las varianzas de emisión, etc. En el caso de Student-t, necesitamos prior para grados de libertad ν si queremos estimarlos; a veces ν se fija para cada estado (p. ej. $\nu=5$) para estabilizar.

Estructura del HMM en notación: $\lambda = { \pi, A, {\mu_j,\Sigma_j,\nu_j}_{j=1..K}}$ donde π es vector de prob inicial, A matriz transiciones, y μ_j,Σ_j,ν_j parámetros de emisión *t* del estado j.

Podemos crear el modelo usando una librería existente (p. ej. `hmmlearn` en Python no soporta Student-t directamente, habría que extenderlo; o usar PyMC/PyStan para especificarlo Bayesianamente). Para propósitos de pseudocódigo, imaginemos una interfaz:

```
# Definir modelo HMM con K estados y emisiones Student-t
model = HiddenMarkovModel(n_states=K, distribution="StudentT")
# Configurar inicialización de parámetros
for j in range(K):
    model.states[j].mu = np.mean(X, axis=0) + np.random.randn(d)*0.1  #
inicializar cerca del promedio global
    model.states[j].Sigma = np.diag(np.var(X, axis=0))  # varianza inicial
aproximada
    model.states[j].nu = 5  # grados de libertad inicial
# Podemos inicializar transiciones uniformemente
model.trans_mat = np.full((K,K), 1.0/K)
```

Si se implementara manualmente, se necesitaría codificar las fórmulas de actualización EM o VB. Una representación de alto nivel del algoritmo VB para entrenar el HMM podría verse así:

```
# Entrenar HMM con inferencia variacional
converged = False
iter = 0
while not converged and iter < max_iter:
    # E-step variacional: computar gamma y xi vía forward-backward
    gamma, xi = forward_backward_var(model, X)
    # (gamma[t,j] = q(z_t=j), xi[t,i,j] = q(z_{t-1}=i, z_t=j))
    # M-step variacional: actualizar parámetros del modelo
```

```
    for j in range(K):
        # actualizar distribución de emisiones del estado j usando gamma[:,j]
como pesos
        Nj = np.sum(gamma[:,j])
        # media:
        model.states[j].mu = np.sum(gamma[:,j].reshape(-1,1) * X, axis=0) / Nj
        # covarianza (asumiendo independencia entre dims por simplicidad):
        diff = X - model.states[j].mu
        model.states[j].Sigma = np.diag(np.sum(gamma[:,j].reshape(-1,
1)*(diff**2), axis=0) / Nj)
        # grados de libertad nu: se puede actualizar por Newton-Raphson o
mantener fijo
        # ...
    # actualizar matriz de transición
    model.trans_mat = np.sum(xi, axis=0)  # sum over t of xi[t,i,j]
    for i in range(K):
        model.trans_mat[i] /= np.sum(model.trans_mat[i])  # normalizar fila i
    # verificar convergencia (p.ej. cambio en log-likelihood < tol)
    loglik = compute_loglikelihood(model, X)
    if abs(loglik - prev_loglik) < tol:
        converged = True
    prev_loglik = loglik
    iter += 1
```

Este pseudocódigo ilustra el flujo: calcular responsabilidades γ y luego re-estimar parámetros ponderando por γ. En inferencia variacional pura, habría pasos adicionales para actualizar hiperparámetros de prior posteriores, pero en la práctica se puede usar *maximum a posteriori* como arriba.

Para un **SSSM**, la definición es más compleja. Si quisiéramos un **Switching AR(1)**, por ejemplo, definiríamos ecuaciones $y_t = c_{z_t} + \phi_{z_t}\,y_{t-1} + \epsilon_{t}$ (modelo de retorno con cambio en intercepto c y autorregresión ϕ según régimen). La inferencia involucraría un filtro de Kalman modificado. Existen librerías especializadas (p. ej. `pyro` / `pyro.contrib.forecast`, `ssm` de Linderman et al.) que soportan estos modelos. No obstante, para muchos casos financieros, un HMM directo sobre retornos (posiblemente incluyéndolos como features retrasadas también) es suficiente para obtener regímenes interpretables (volátil vs tranquilo, etc.).

5.3 Inferencia y ajuste del modelo

Tras definir el modelo, procedemos a estimar sus parámetros con los datos históricos. Usaremos el algoritmo de inferencia seleccionado (sea EM tradicional o VB). Vale la pena dividir los datos en un período de entrenamiento (in-sample) y luego validar out-of-sample. Por ejemplo, entrenar el HMM con datos de 2000–2015, y luego usar 2016–2020 para evaluar.

Durante la inferencia es importante monitorear la convergencia. En HMM, comúnmente se observa la **log-verosimilitud** $ \log P(y_{1:T}|\text{modelo})$ en cada iteración EM/VB para ver si se estabiliza. Si el entrenamiento es bayesiano completo (por ejemplo usando MCMC), se debe chequear la convergencia de la cadena y la mezcla de estados.

Una vez entrenado el modelo, podemos inspeccionar los **parámetros aprendidos** para interpretar los regímenes: por ejemplo, la media de retornos μ_j y volatilidad Σ_j de cada estado j. Supongamos que el modelo resultante tiene 3 estados con parámetros estimados como en la siguiente tabla hipotética:

- Estado 0: $\mu_0 \approx +0.1\%$ de retorno diario, varianza de retorno $\approx (0.5\%)^2$, volumen medio normal. (Régimen alcista calmado).
- Estado 1: $\mu_1 \approx -0.05\%$, varianza $\approx (1\%)^2$, volumen moderado. (Régimen neutral/ligeramente bajista con algo de volatilidad).
- Estado 2: $\mu_2 \approx -0.3\%$, varianza $\approx (2\%)^2$, volumen muy alto. (Régimen de crisis – caídas fuertes con altísima volatilidad).

Estos parámetros ya dan intuición económica. Además, la **matriz de transición** A dirá la persistencia: por ejemplo, si $a_{00}=0.95$ significa que el régimen alcista calmado suele durar bastante (una vez en él, 95% chance de seguir al día siguiente), mientras que si $a_{02}=0.10$ indica que aunque raro, hay ~10% de chance de saltar de alcista calmado directamente a crisis (quizás asociado a shocks). A menudo, visualizamos A como grafo dirigido con probabilidades para entender el *ciclo* de regímenes (algunos modelos encuentran ciclos recurrentes, otros transiciones libres).

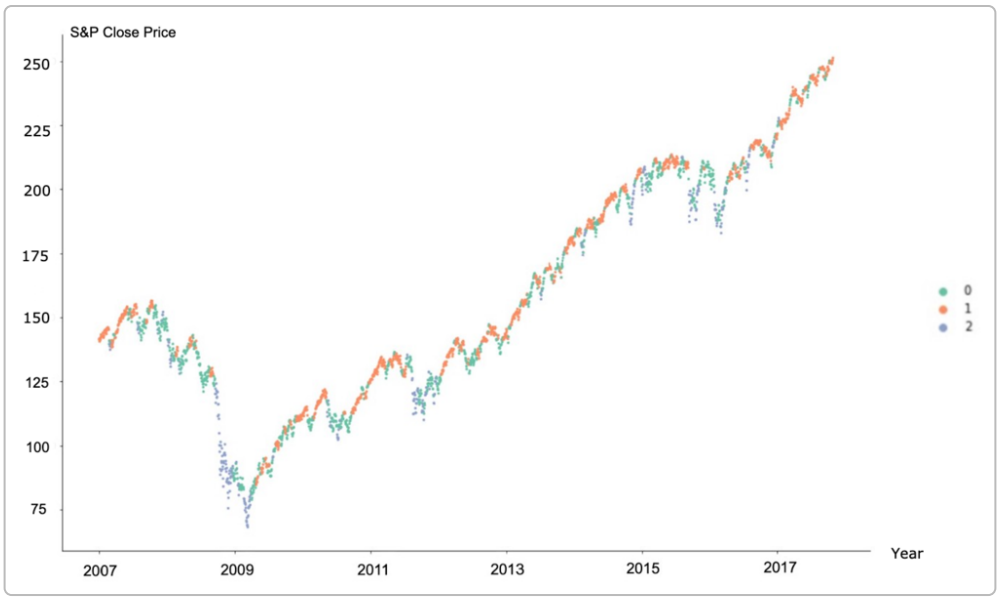
5.4 Detección de cambios de régimen (decodificación de estados ocultos)

Una vez ajustado el modelo, aplicamos **decodificación** a la serie temporal para obtener la secuencia de estados ocultos más probable $z_{1:T}$ o las probabilidades filtradas de cada estado en cada instante. Existen dos enfoques principales:

- **Decodificación *smoothing*** (posterior completa): calcula $P(z_t = j \mid y_{1:T})$ para cada t,j usando el algoritmo forward-backward una vez finalizado el entrenamiento. Esto brinda las probabilidades *a posteriori* suavizadas, utilizando *toda* la serie. Podemos asignar a cada tiempo el estado de máxima probabilidad ("decodificación MAP global").
- **Decodificación *online*** (filtro): calcula $P(z_t = j \mid y_{1:t})$ recursivamente al avanzar t. Esto sería útil en tiempo real (ej. para generar señales al vuelo), aunque es menos preciso que el smoothing retrospectivo porque no usa datos futuros. A posteriori, normalmente preferimos smoothing para análisis.

Un algoritmo clásico es el de **Viterbi**, que encuentra la secuencia de estados ocultos más verosímil en conjunto (maximiza $P(z_{1:T}\mid y_{1:T})$). El Viterbi produce una secuencia única, que puede diferir ligeramente de la secuencia obtenida escogiendo estado de máxima probabilidad marginal en cada t; sin embargo, en prácticas financieras a menudo las diferencias son pequeñas y se suele reportar la secuencia Viterbi para simplicidad.

Identificación de cambios de régimen: Una vez tenemos la secuencia estimada de estados $z_{1:T}$, detectamos los puntos de cambio donde $z_t \neq z_{t-1}$. Estas fechas son candidatos a **switches estructurales** en el mercado. Por ejemplo, el modelo podría inferir que el estado cambió de 0 a 2 el 15/Sep/2008 (Lehman Brothers) indicando entrada abrupta en régimen de crisis, luego cambió a 1 en 2009 tras estabilización, etc. La Figura 4 muestra una serie de precios del S&P500 coloreada según el régimen inferido (3 estados)



. Se observan claramente segmentos de distinto color: por ejemplo, 2007–2009 la mayoría es azul (estado 2, crisis), 2013–2014 mayoritariamente naranja (estado 0, alcista volatilidad baja), etc. Estas visualizaciones ayudan a validar que los regímenes hallados corresponden a intuiciones históricas.

También podemos computar la **duración media** estimada de cada régimen: $\mathbb{E}[\text{duración}*jj* = *1/(1 - a*=0.93$ (crisis dura ~14 días). Esto concuerda con la intuición de que mercados neutrales son transición relativamente breve, mientras fases definidas alcistas o de crisis duran más. Es útil señalar cada }) $. Si a_{jj} está cerca de 1, la duración esperada es alta. En nuestro ejemplo hipotético, supongamos $a_{00}=0.97$ (régimen alcista dura en media ~33 días), $a_{11}=0.80$ (neutral dura ~5 días), $a_{22*run* (racha) de régimen en la serie – estas rachas marcan los períodos que un analista podría calificar como "regímenes" en retrospectiva.

5.5 Generación de señales estructurales a partir del modelo

Los regímenes ocultos estimados pueden emplearse para generar **señales operativas** de varias formas:

- **Filtro de riesgo macro:** Como en el ejemplo de QuantStart [45] [46], se puede usar el estado oculto como un **"overlay" de riesgo** sobre otra estrategia. Por ejemplo, permitir posiciones largas solo si el régimen actual es de volatilidad baja (favorable), y desactivar o reducir exposición si el modelo indica un régimen adverso. En nuestro caso, podríamos decir: si $P(z_t=\text{crisis})>0.6$, entonces elevar liquidez o coberturas (señal de *risk-off*). En cambio, si $P(z_t=\text{bull})$ es alta, aumentar exposición a renta variable (*risk-on*). Estas señales no predicen direccionalmente el mercado en el corto plazo, pero ayudan en la **asignación táctica de activos** y en gestión de riesgo (p. ej., dimensionamiento de posiciones según el régimen).
- **Señales de cambio de tendencia:** Algunas estrategias buscan anticipar o reaccionar rápido a un cambio de régimen. Si el modelo detecta un cambio $z_{t-1} \to z_t$ significativo (por ejemplo de bull a bear), eso puede disparar una señal (vender acciones, comprar bonos, etc.). Sin embargo, hay que considerar el **retardo**: los HMM detectan cambios una vez ocurridos unos cuantos pasos de datos en el nuevo régimen, para tener suficiente evidencia. Usar *smoothing* implica ver el cambio con retrospectiva. Para trading *online*, conviene usar la probabilidad filtrada $P(z_t=j \mid y_{1:t})$ e

interpretar un aumento rápido de probabilidad de un régimen distinto como alerta. En pseudocódigo:

```
state_prob = forward_filter(model, X_live)  # obtener P(z_t | datos hasta t) en tiempo real
if state_prob[t]['crisis'] > 0.7 and state_prob[t-1]['crisis'] < 0.5:
    signal = "ALERTA: posible cambio a régimen de crisis en día t"
```

Esta señal podría complementarse con otras (p. ej., ruptura de soportes técnicos) para confirmar acción. - **Construcción de portafolios robustos a regímenes:** Otra aplicación es generar **señales de rotación de factores** o activos. Por ejemplo, un estudio (Wang et al. 2020) aplicó HMM para alternar entre modelos de factor investing dependiendo del régimen detectado [47] [48]. Los estados ocultos servían como señal para *conmutar* entre un portafolio momentum vs uno valor, logrando así mejorar desempeño. En general, las señales estructurales pueden ser: "estar en acciones vs liquidez", "rotar de sectores cíclicos a defensivos", "cambiar parámetros de un algoritmo (p. ej., el horizonte de un sistema de trading)", todo ello disparado por el indicador de régimen. - **Indicadores informativos:** A veces la señal no es una instrucción directa de trading sino una métrica de riesgo. Por ejemplo, crear un **"índice de régimen"** en [0,1] que combine las probabilidades de estar en regímenes extremos. Un gestor podría reportar: "El modelo de regímenes indica un 80% de probabilidad de estar en ambiente de alta volatilidad; riesgo sistémico elevado". Esto ayuda en comunicación y decisiones discrecionales.

En cualquier caso, es esencial **validar las señales**. Un backtesting sencillo es simular una estrategia que sigue las señales de régimen y comparar contra benchmark. Por ejemplo, estrategia: invertir en S&P500 solo en régimen bull, y en bonos del tesoro en régimen bear. Evaluamos CAGR, Sharpe, drawdowns. Si la identificación de regímenes tiene valor predictivo (o al menos evita pérdidas grandes), deberían verse métricas mejoradas frente a buy-and-hold. En Table 4 de Wang et al. (2020) se ven mejoras significativas en Sharpe y drawdown al incluir la dinámica de regímenes [49] [50].

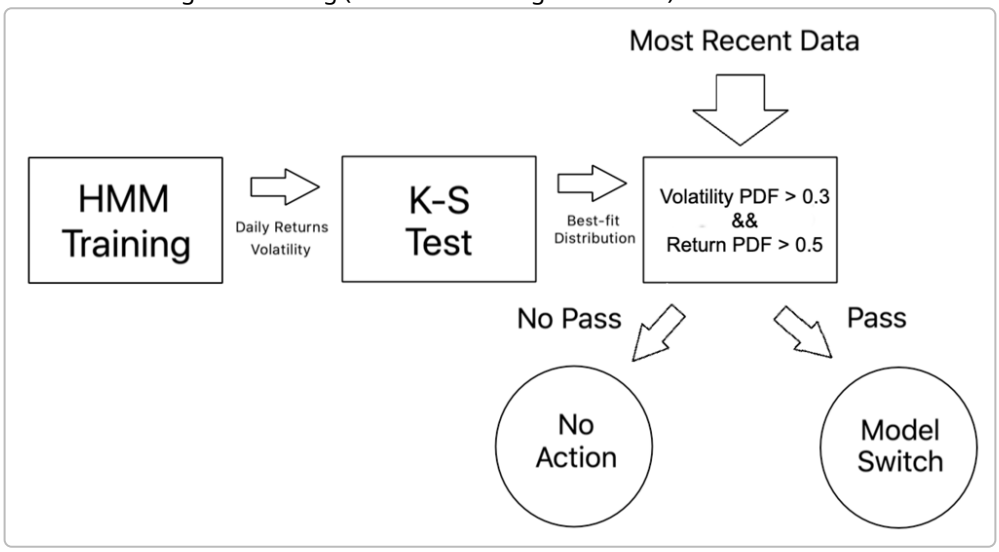
5.6 Evaluación del desempeño del modelo y las señales

La evaluación ocurre en dos niveles: **(1) Desempeño del modelo de régimen** en sí, y **(2) Desempeño de las señales derivadas** en la aplicación (trading, gestión riesgo).

Para (1), podemos mirar: - **Log-verosimilitud fuera de muestra:** ¿Qué tan bien el modelo predice la distribución de retornos en periodos no usados en entrenamiento? Esto se mide con $\log P(y_{\text{test}} \mid \text{modelo})$. Un modelo de régimen bueno típicamente supera a un modelo simple (e.g. GARCH o Normal i.i.d.) en explicar colas y clustering de volatilidad. - **Calidad de clasificación de eventos históricos:** Si tenemos "ground truth" aproximado (p. ej. sabemos fechas de crisis), ver cuántas las detectó el modelo. No es supervisado, así que no es justo exigir correspondencia 1 a 1, pero cualitativamente podemos revisar si los regímenes corresponden a períodos conocidos (crisis 2008, COVID crash 2020, etc.). En nuestro ejemplo, el régimen 2 debía encender en 2008 y 2020; si el modelo en cambio creó un régimen separado para 2011 (crisis europea) y no mezcló con 2008, puede interpretarse. - **Estabilidad y interpretabilidad:** ¿Los parámetros estimados tienen sentido económico? ¿Se parecen los regímenes si repetimos el entrenamiento con datos ligeramente diferentes (robustez)? Modelos extremadamente sensibles pueden no ser confiables.

Para (2) (la aplicación), la evaluación es mediante **backtesting** o simulación histórica: - Aplicar las reglas de señal a la serie histórica y calcular métricas: rendimiento acumulado, Sharpe ratio, máximo drawdown, volatilidad, ratio de información vs benchmark, etc. Por ejemplo, si la estrategia con filtro de régimen incrementa el Sharpe de 1.0 a 1.5 y reduce drawdown de 50% a 30%, son señales de valor agregado [51]. En Table 4 y 5 de Wang et al., se ven las métricas de un modelo con HMM vs varios benchmarks [49] [52]. - Test de estabilidad: dividir el periodo test en subperíodos, ver si la estrategia funciona consistentemente (no solo por calibrar bien una crisis). - Analizar *timing* de las señales: ¿El modelo cambia a régimen bajista *después* de una caída de 30% (útil pero tardío) o logramos detectar tras 5% de caída? ¿Cuántos falsos positivos (cambios a crisis que luego revertían rápido)? Esto importa para implementabilidad real. - Considerar costos de transacción: si la estrategia produce señales muy frecuentes de entrar/salir, quizás los beneficios se evaporan con costes. Un buen modelo de regímenes normalmente no cambia cada día – ofrece estabilidad subyacente. Por eso añadir la componente "sticky" es valioso, para no reaccionar a cada ruido pequeño.

Resumen de pipeline: Ingerir OHLCV → derivar features (retorno, volatilidad, etc.) → entrenar HMM/SSSM robusto (posiblemente con Student-t y variacional) → obtener secuencia de regímenes → extraer de ahí información útil (cambios, probabilidades) → accionar una estrategia o alerta → medir resultados. Este proceso puede automatizarse y actualizarse periódicamente (por ejemplo, re-entrenar el modelo cada mes con datos hasta la fecha para incorporar nueva información, aunque cuidado de datos futuros en forward-looking). La **Figura 5** muestra esquemáticamente un diagrama de flujo para el uso de un modelo de regímenes en una estrategia de trading (tomado de Wang et al. 2020)



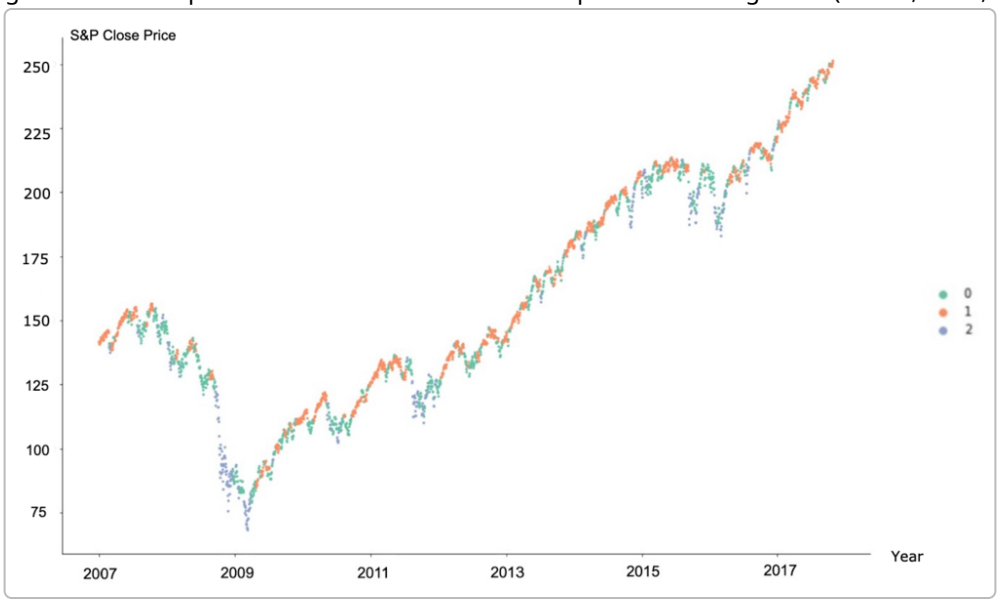
. En él, el modelo HMM entrenado produce una distribución que se contrasta con datos recientes (prueba K-S) para decidir si se está en un régimen particular, lo cual desencadena una acción (o "No Action"). Si bien este flujo específico incluye un test estadístico adicional, conceptualmente ilustra cómo las salidas del modelo (distribución de volatilidad y retorno) alimentan reglas de decisión de inversión.

6. Conclusiones

Los modelos HMM y SSSM avanzados proporcionan un marco matemático sólido para identificar cambios de régimen en series financieras, capturando fenómenos de **volatilidad en racimos**, **colas pesadas** y **cambios estructurales** que caracterizan a los mercados. La incorporación de distribuciones *Student-t* en las

emisiones aporta realismo al modelar retornos financieros con eventos extremos [5]. El uso de inferencia variacional en lugar de EM clásico mejora la escalabilidad y estabilidad del ajuste, a la vez que habilita la selección automática de complejidad del modelo (número de estados) evitando sobreajuste [17]. Métodos no paramétricos como HDP-HMM llevan esto al límite, permitiendo un número potencialmente infinito de regímenes y aprendiendo efectivamente cuántos son necesarios a partir de los datos [28] [29]. Además, hemos discutido cómo enriquecer los modelos con información macroeconómica de manera suave – lo que podríamos denominar **HMM híbridos** – para guiar la detección de regímenes conforme al contexto económico sin perder la naturaleza no supervisada del aprendizaje [41] [42].

La aplicación práctica a datos OHLCV demuestra que estos modelos pueden integrarse en un pipeline analítico completo: desde la extracción de *features* relevantes (retornos, volatilidad, volumen) hasta la generación de señales operativas de *trading* o *asset allocation*. Los ejemplos y visualizaciones presentados (Figuras 1–5) ayudan a entender la arquitectura del modelo y el flujo de datos: se observó cómo los estados ocultos segmentan el comportamiento del índice S&P500 en períodos distinguibles (alcista, crisis, etc.)



, y cómo dichas segmentaciones pueden usarse para mejorar decisiones de inversión. El pseudocódigo provisto clarifica la implementación de cada componente clave en estilo Python, lo cual debería facilitar a un equipo cuantitativo reproducer estas técnicas con sus propios datos.

En conclusión, las técnicas modernas de modelado de cambios de régimen combinando HMM/SSSM con inferencia Bayesiana y conocimientos macro representan una poderosa herramienta en el arsenal de la ingeniería financiera. Permiten **quantificar y adaptarse** a las fases cambiantes del mercado de manera probabilística y dinámica. Para una audiencia técnica, los beneficios son claros: modelos más realistas (colas pesadas, persistencia de regímenes), inferencia más robusta (variacional, no paramétrica) y mejores integraciones con señales de mercado existentes. La literatura académica respalda estos avances – desde Rabiner (1989) [53] en HMM básicos, pasando por Hamilton (1989) en econometría de regímenes, hasta trabajos recientes como Fox (2011) en HDP-HMM [30] o Chatzis (2011) en HMM-t variacional [17]. Implementados correctamente, estos modelos ofrecen una **ventana estadística** para entender la volatilidad y riesgo financiero, y una base cuantitativa para estrategias adaptativas que puedan navegar por regímenes de mercado dispares con mayor eficacia.

Referencias Seleccionadas:

- Rabiner, L. R. (1989). *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*. Proc. IEEE, 77(2). (Referente clásico de HMM) [53] [54].
- Hamilton, J. D. (1989). *A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle*. Econometrica, 57(2). (Modelo de cambio de régimen Markoviano en macroeconomía).
- Chatzis, S., Kosmopoulos, D., & Varvarigou, T. (2010). *A Variational Bayesian Methodology for Hidden Markov Models utilizing Student's-t Mixtures*. Pattern Recognition 44(2):295-306. (HMM de Student-t, inferencia variacional) [17] [18].
- Fox, E. B., et al. (2011). *A Sticky HDP-HMM with application to Speaker Diarization*. Annals of Applied Statistics 5(2A):1020-1056. (Introducción del HDP-HMM "sticky" para evitar cambios excesivos) [30] [31].
- Wang, M. et al. (2020). *Regime-Switching Factor Investing with Hidden Markov Models*. J. Risk Financial Manag. 13(12), 311. (Aplicación práctica de HMM a inversión por factores, con visualizaciones) [47] [55].
- Wang, E. T. et al. (2023). *Bayesian Non-homogeneous Hidden Markov Model with Variable Selection for Investigating Drivers of Seizure Risk Cycling*. Ann. Appl. Stat. 17(1):333-356. (Ejemplo de HMM con covariables externas en transiciones/emisiones) [41] [42].

---

[1] [2] [3] [4] Modelo_oculto_de_Markov
https://www.quimica.es/enciclopedia/Modelo_oculto_de_Markov.html

[5] [7] [8] [9] [10] scitepress.org
https://www.scitepress.org/Papers/2021/101506/101506.pdf

[6] (PDF) Student's t-Hidden Markov Model for Unsupe rvised Learning ...
https://www.researchgate.net/publication/318760285_Student's_t-Hidden_Markov_Model_for_Unsupe_rvised_Learning_Using_Localized_Feature_Selection

[11] [12] [13] [25] [26] cs.toronto.edu
https://www.cs.toronto.edu/~hinton/absps/switch.pdf

[14] [15] Switching Kalman Filters for Prediction and Tracking in an Adaptive Meteorological Sensing Network
https://cecs.uci.edu/~papers/secon05/DATA/05-03.PDF

[16] Hidden Markov model - Wikipedia
https://en.wikipedia.org/wiki/Hidden_Markov_model

[17] [18] [19] [20] [21] [22] [23] [24] A variational Bayesian methodology for hidden Markov models utilizing Student's-t mixtures | Request PDF
https://www.researchgate.net/publication/222298489_A_variational_Bayesian_methodology_for_hidden_Markov_models_utilizing_Student's-t_mixtures

[27] [30] [31] [32] [33] (PDF) A sticky HDP-HMM with application to speaker diarization
https://www.researchgate.net/publication/45851769_A_sticky_HDP-HMM_with_application_to_speaker_diarization

[28] [PDF] Speech Acoustic Unit Segmentation Using Hierarchical Dirichlet ...
https://www.isca-archive.org/interspeech_2013/torbati13_interspeech.pdf

[29] [PDF] Action Classification using a discriminative multilevel HDP-HMM
https://core.ac.uk/download/pdf/42132872.pdf

34 35 Illustrative example of a Switching Linear Dynamical System (SLDS). The... | Download Scientific Diagram
https://www.researchgate.net/figure/Illustrative-example-of-a-Switching-Linear-Dynamical-System-SLDS-The-stochastic_fig1_270454498

36 37 47 48 49 50 51 52 53 54 55 Regime-Switching Factor Investing with Hidden Markov Models
https://www.mdpi.com/1911-8074/13/12/311

38 39 40 41 42 43 44 BAYESIAN NON-HOMOGENEOUS HIDDEN MARKOV MODEL WITH VARIABLE SELECTION FOR INVESTIGATING DRIVERS OF SEIZURE RISK CYCLING - PMC
https://pmc.ncbi.nlm.nih.gov/articles/PMC10939012/

45 46 Market Regime Detection using Hidden Markov Models in QSTrader | QuantStart
https://www.quantstart.com/articles/market-regime-detection-using-hidden-markov-models-in-qstrader/