Mauricio Ferreira de Araujo - 222007021
Disciplina: Inteligência Artificial

Portfólio 3 - Modelos Markovianos ocultos

1 Introdução

Este projeto tem como principal objetivo apresentar a explicação do problema e o exemplo de uso do algoritmo desenvolvido utilizando modelos Markovianos ocultos. Este material é referente ao Portfólio 3 da disciplina de inteligencia artificial. Além da descrição do problema e da demonstração do algoritmo, serão também apresentadas imagens da execução do código, servindo como comprovação visual do seu funcionamento e dos resultados obtidos.

2 Problema

No mercado, o preço de uma ação é apenas a observação superficial de um processo que é invisível e incerto. As condições reais do mercado como se ele está em um estado de "alta volatilidade e queda", "baixa volatilidade e estabilidade", ou "alta volatilidade e alta"não são diretamente observáveis, elas são os estados ocultos, como não temos informação completa sobre qual regime o mercado está operando em um determinado dia.

O desafio fundamental é, portanto, como conseguimos modelar e tomar decisões racionais ao inferir a sequência mais provável desses estados ocultos apenas observando as variáveis ruidosas.

Além disso, a complicação é que esses estados ocultos não são independentes, eles têm uma dinâmica de transição. O estado de hoje influencia a probabilidade do estado de amanhã. O problema, então, é conseguir representar de forma compacta e eficiente esse conhecimento de domínio, inferindo a probabilidade de transição entre os regimes e qual distribuição estatística cada regime gera nas observações. O objetivo final é desvendar a estrutura escondida que gera os dados de mercado que vemos.

3 Projeto utilizando conceito de Modelos Markovianos ocultos para solução de um problema

Começamos dando um chute inicial. O código chuta onde estão os regimes e chuta qual a probabilidade de começar em cada um e de transitar entre eles.

Aí entra o treinamento, que é o Baum-Welch (algoritmo EM), rodando em loop por várias repetições.

1. Passo E (Expectation - Cálculo das Probabilidades): O código pega todos os dados observados e tenta descobrir qual é a chance do mercado ter estado em cada um dos regimes ocultos em cada dia. Essa etapa gera as probabilidades de estado para cada dia.

2. Passo M (Maximization): Agora que o código tem uma ideia de onde o mercado "provavelmente"esteve, ele usa essas probabilidades como pesos para ajustar os parâmetros do modelo. Ele recalcula:

    - A probabilidade de transição: Se, na prática, ele viu o mercado sair muito de um regime para outro, a probabilidade de transição é corrigida para refletir essa mudança.
    - As médias e covariâncias de cada estado: Se um regime foi mais associado a dias de volatilidade realmente baixa, o valor médio e a covariância desse regime são ajustados para focar mais nessa característica.

Esse processo se repete até que os ajustes sejam mínimos, ou seja, o modelo converge e encontrou a estrutura de regimes mais provável que gerou os dados de mercado.

Finalmente, depois de treinado, o algoritmo Viterbi entra em ação. O Viterbi pega o modelo ajustado e, para cada dia, não calcula a probabilidade, mas sim a sequência mais provável de estados ocultos que o mercado percorreu. Ele transforma a série de preços ruidosa em uma série de etiquetas que dizem: "Neste período, o mercado estava no Regime 1, depois mudou para o Regime 3". Isso resolve o problema de inferir o estado oculto que estava no comando.

3.1 Exemplo de uso

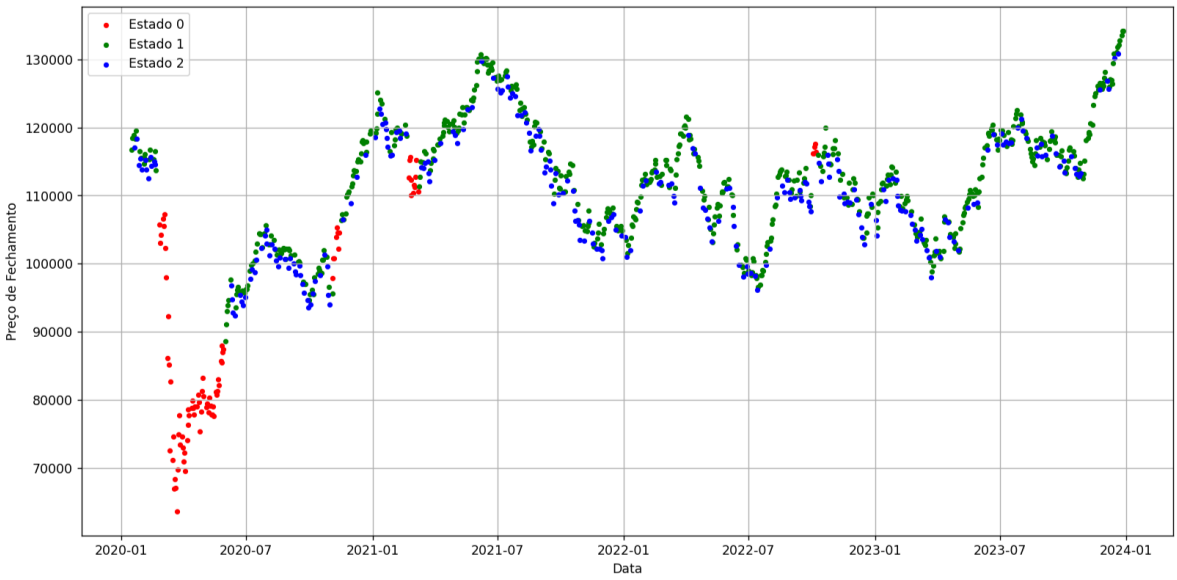

Figura 1: Execução do algoritmo desenvolvido

4 Reflexão pessoal sobre o processo de desenvolvimento

Sendo o mercado financeiro um tema que eu gosto muito, ter a oportunidade de pesquisar sobre o mesmo em uma materia da faculdade é bem motivador, porque se torna algo divertido e não somente uma tarefa aleatoria de uma materia, alem disso, estudar esses

modelos markovianos ocultos faz com que eu fique pensando se esse é o melhor jeito de usar ele ou daria pra explorar mais os limites.

5 Desafios encontrados

Minha ideia inicial era implementar esse modelo usando dados reais. Eu até consegui achar a biblioteca certa em Python para fazer a coleta, mas o desafio mesmo foi entender e processar o que ela retornava. Foi bem complicado adequar esse output complexo ao formato que o algoritmo precisava para rodar e, assim, conseguir os resultados que eu esperava.