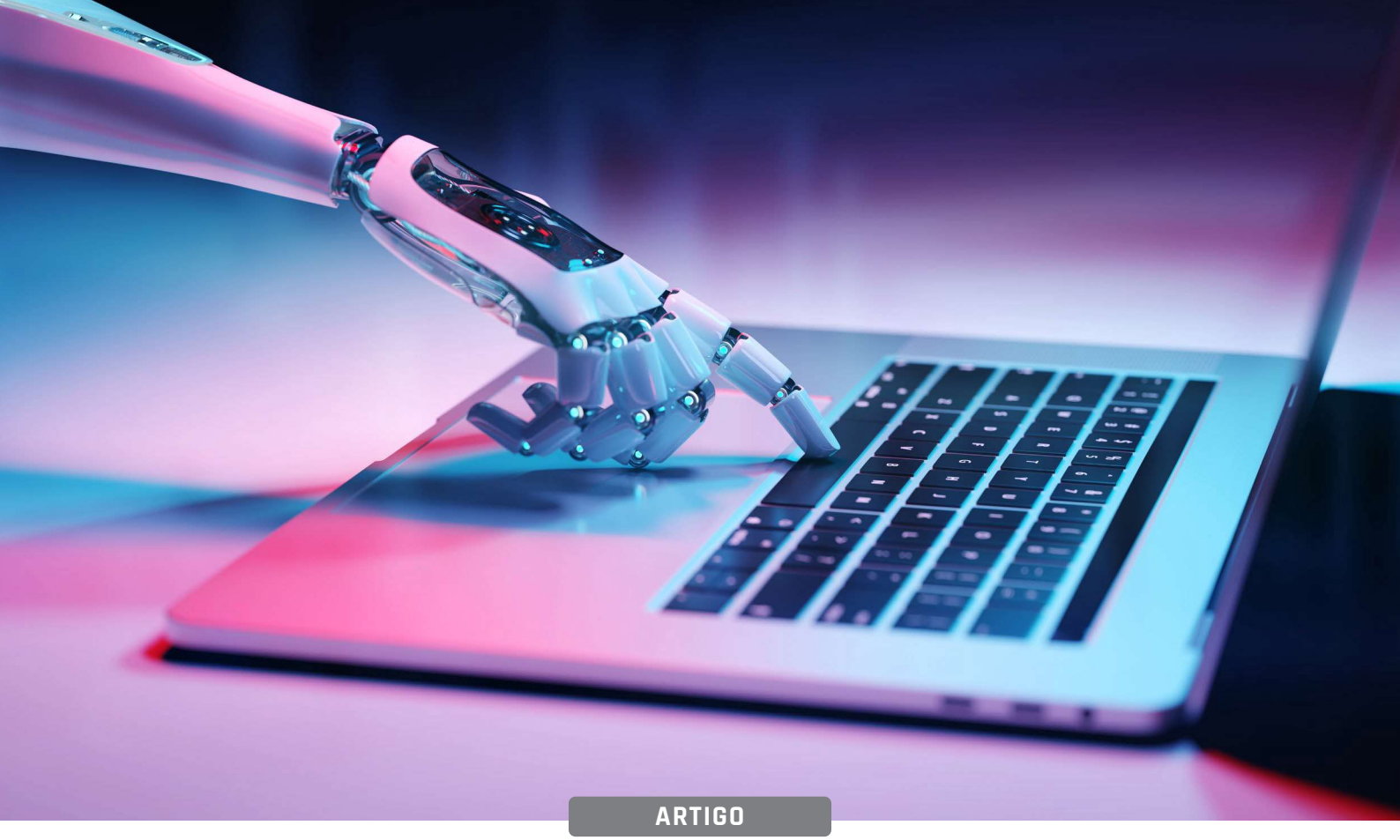


ARTIGO

MODELAGEM DE SISTEMAS DE COMPUTAÇÃO E SUA CONEXÃO COM APRENDIZADO DE MÁQUINA BASEADO EM MODELOS

POR

Edmundo de Souza e Silva

edmundo@cos.ufrj.br

O mundo está repleto de exemplos onde objetivamos prever o comportamento de sistemas, como sistemas de computação, saúde de um grupo de indivíduos ou de uma pessoa (afinal somos um sistema complexo), Internet e mercado de ações, para citar alguns. Previsões sobre como um sistema responderia a mudanças na sua estrutura ou no ambiente em torno são comuns. Modelagem e análise de sistemas são elementos essenciais no projeto da maioria dos sistemas nas engenharias e, em particular, nos sistemas de computação.

Por exemplo, quando um novo protocolo de rede de computadores é projetado, geralmente o seu desenvolvimento vem acompanhado de um modelo que possibilite mostrar as vantagens sobre os protocolos existentes, e em quais condições o novo protocolo poderia operar de forma eficiente. Evidentemente, é preciso definir o que significa "operação eficiente". Analogamente, é importante tentar prever o comportamento de algoritmos, tais como os escalonadores de tarefas, algoritmos dis-

tribuídos ou de resolução de contenção; tentar estimar a eficiência ou a confiabilidade de novas arquiteturas de sistemas de computação; antecipar o comportamento de clientes ao acessar um site; e fazer análise de *trade-offs* num processo de tomada de decisão. Esses são apenas poucos exemplos dentre os inúmeros onde a modelagem desempenha um papel fundamental na computação.

O termo "modelo" refere-se a uma representação de um sistema, que se espera ser precisa e nos ajude a entender o seu comportamento. Um modelo pode ser uma estrutura física que imita o comportamento do sistema, em pequena escala, ou um conjunto de equações cuja solução permite o cálculo das métricas que procuramos estimar, dadas as condições de entrada. Independentemente da representação, modelos devem ser fáceis de compreender, devem representar o sistema com precisão e sem nenhum viés pessoal. Devem também fornecer alguma intuição sobre o comportamento do sistema e auxiliar na definição dos objetivos que se deseja alcançar e na escolha das ações apropriadas para atingi-los. Todos os modelos são apenas aproximações de um sistema real e simplicidade e flexibilidade são atributos muito importantes para a utilidade de um modelo. Quando o sistema a ser avaliado já existe, o analista, usando sua experiência, infere os principais atributos que influenciam as métricas a serem estimadas. A partir dessa abstração mental, um modelo matemático é construído.

Para tornar a discussão mais concreta, consideremos um exemplo simples e fácil de compreender, mas que foi muito útil na análise de sistemas computacionais. Provavelmente, todos nós já vivenciamos a desagradável situação de ter que esperar em uma longa fila para obter algum tipo de serviço, por exemplo, para falar com um atendente de banco ou para obter ajuda de um *call center*. Portanto, prever o tempo de espera nas filas é importante. Para despertar o interesse do leitor da área de computação, vale destacar que filas estão presentes em todos os lugares! Em um computador, existem inúmeras filas, como as filas de acesso a uma CPU para executar instruções e processar dados; buffers e caches nas hierarquias de memória; filas de E/S para diferentes dispositivos; filas do sistema operacional para *threads* a espera de escalonamento; filas de pacotes para a conexão com a internet, e assim por diante.

Como é mais fácil imaginar uma fila formada por pessoas que requerem algum serviço do gerente do banco, vamos usar o exemplo de fila única e com um único atendente, e supor que precisamos estimar o tempo médio que uma pessoa terá de esperar até conseguir falar com o gerente, digamos, para pedir um empréstimo. Como seria possível construir um modelo para este exemplo trivial? Para responder a essa pergunta, instigamos a intuição do leitor que poderia supor que será necessário estimar o tempo entre a chegada de cada cliente e o tempo que um cliente ocupa o gerente. Essas duas quantidades são variáveis aleatórias, pois o intervalo

entre chegadas de clientes varia e, provavelmente, cada cliente ocupará o gerente por um tempo diferente do outro. Como essas são variáveis aleatórias, também é preciso estimar a distribuição de cada uma. Dependendo das escolhas que fizermos para essas distribuições, precisaremos apenas de seus valores médios, e então poderemos construir um modelo de Markov e obter expressões para o tempo médio que os clientes esperam na fila.

A noção de estado é outro conceito essencial. A escolha apropriada do espaço de estados geralmente depende da aplicação e do que queremos estimar e é um dos aspectos importantes na construção de um modelo. Neste exemplo simples, o estado é o número de clientes na fila. Com o nosso modelo, é possível estimar o número de médio de clientes em espera após um determinado intervalo de tempo. Incentivamos o leitor a refletir sobre como obter as seguintes medidas de interesse: as variáveis necessárias: o tempo médio que um cliente espera na fila; o tempo médio necessário para esvaziar a fila, a partir do instante em que existe um número específico de clientes no sistema; e obter respostas a outras possíveis perguntas que seriam interessantes.

Uma importante parte do processo de modelagem é a execução de experimentos e obtenção de métricas pertinentes ao que se deseja analisar [1]. Tais métricas auxiliam no ajuste dos parâmetros do modelo e na validação dos resultados fornecidos pelo mesmo. Ao mesmo tempo, um modelo também é útil para orientar o processo de experimentação e para verificar os resultados dos experimentos, identificando erros no processo de medição ou algum viés involuntário nos resultados. Caso uma métrica observada seja muito diferente da obtida pelo modelo, ou o modelo é impreciso ou algo está errado com o procedimento de coleta da métrica. Geralmente há um modelo mental subjacente a qualquer processo de medição e, caso o sistema real exista, experimentação e modelagem devem andar de mãos dadas.

Quando o analista finalmente adquirir confiança no modelo, este pode ser usado para prever comportamentos futuros ou para responder perguntas do tipo "what-if". Por exemplo, qual seria o impacto no desempenho do sistema caso um valor específico de um parâmetro de entrada exceda um certo percentual? Um modelo também pode ser usado para determinar as condições sob as quais o sistema alcançaria os objetivos especificados. Como exemplo, um analista poderia avaliar se uma arquitetura atende aos requisitos de confiabilidade do sistema e, caso não atinja, identificar os componentes principais do sistema que afetam adversamente a sua confiabilidade.

Caso o sistema não tenha sido construído, talvez por estar na fase de projeto, um modelo é crucial para comparar alternativas possíveis, com diferentes arquiteturas ou algoritmos a serem usados, além de auxiliar na fase de configuração. Várias perguntas podem ser respondidas tais como "quantos servidores precisamos usar em um cluster de computadores para processar a carga de trabalho esperada e mantendo o tempo de resposta abaixo de um dado limite?" ou ainda, "qual é a pro-

babilidade de um sistema falhar durante seu tempo de vida útil?".

Modelos de filas ou de redes de filas [2], juntamente com modelos de Markov, têm sido amplamente utilizados ao longo dos anos. Além dos exemplos já mencionados, esses modelos têm sido recentemente empregados no projeto de datacenters. A internet como a conhecemos hoje é outro exemplo notável onde modelos fundamentais foram empregados desde a sua criação [3]. Isso inclui o trabalho do vencedor do Prêmio Turing de 2022 pela invenção da Ethernet, cuja eficiência do protocolo (CSMA/CD) foi demostrada por um modelo de Markov. O uso de modelos deste tipo continua com trabalhos recentes de redes quânticas, dentre vários outros.

Modelos também são úteis para determinar ações que otimizem alguma métrica específica de recompensa [4]. Evidentemente, devemos definir o que entendemos por "recompensa". Uma recompensa pode ser uma medida de ganho monetário, de eficiência ou qualquer outra métrica que interesse ao projetista.

Nós humanos estamos constantemente tomando decisões. Ao comprar um carro usado, avaliamos seu preço atual, condição mecânica, idade, aparência, número de proprietários anteriores, etc. Qual recompensa estamos procurando? Ela poderia ser uma função da expectativa de vida útil do carro, da frequência de reparos futuros esperada ou da satisfação que a compra nos trará. Note que procuramos verificar o estado do carro (condição mecânica, etc.), juntamente com outras variáveis de entrada (preço, número de proprietários anteriores), e então avaliamos a recompensa que o carro nos trará. Antes de tomar uma decisão, construímos um modelo mental baseado em nossas experiências passadas. Alternativamente, podemos confiar no julgamento de especialistas que já avaliaram carros similares sob condições semelhantes.

A maioria dos sistemas reais é extremamente complexa e realizar uma representação detalhada em um modelo é muito difícil, se não impossível. Portanto, a chave para entender um sistema complexo é buscar seu comportamento macroscópico, pois conhecer os detalhes de cada componente não necessariamente fornece qualquer pista para prever o comportamento do sistema com um todo.

Agora que o leitor deve estar familiarizado com a Modelagem de Sistemas de Computação (CSM em inglês), cabe a pergunta: qual seria a conexão entre CSM e Aprendizado de Máquina baseado em Modelos ("*model-based Machine Learning*"-*ML*)? Primeiro, precisamos definir *Machine Learnin*g. ML pode ser definido em linhas gerais como um conjunto de métodos que possibilitam a detecção automática de padrões em dados" [6]. Ou como um conjunto de "métodos que são úteis para fazer os sistemas de computação realizarem uma tarefa cada vez melhor através da experiência" [7]. Outra definição seria "um processo que permite aos computadores aprenderem com os dados e melhorarem seu desempenho ao longo do tempo sem serem explicitamente programados" [8]. Seme-

lhante a CSM, usamos em ML um modelo (estatístico) para representar o sistema real que queremos estudar.

Uma distinção fundamental entre CSM e ML é que, no caso do ML, após a escolha de um modelo para representar o mundo real, os parâmetros desse modelo são "aprendidos" a partir dos dados disponíveis. Atualmente, produzimos uma vasta quantidade de dados e a capacidade computacional ao nosso dispor é suficientemente grande para "aprender" os parâmetros de um modelo.

Concentremos nosso foco no aprendizado por reforço (*Reinforcement Learning - RL*). No RL, um agente observa o estado atual do ambiente e toma uma ação, a partir das escolhas possíveis disponíveis naquele estado. O agente recebe uma recompensa ou penalidade após realizar uma ação. Com o passar do tempo, o agente aprende a melhor decisão (aprende uma política) que otimiza uma função das recompensas recebidas. Note que, como no CSM, devemos estar familiarizados com a noção de estado e como ele muda ao longo do tempo. No CSM, o estado é alterado de acordo com eventos aleatórios (por exemplo, a chegada ou a partida de um cliente), enquanto no RL, o estado muda de acordo com a escolha da ação que o agente toma. O formalismo padrão usado para definir problemas de RL é chamado de Processo de Decisão Markoviano (MDP), que é baseado em modelos de Markov, a ferramenta básica para CSM.

Tanto o RL quanto o CSM utilizam modelos estocásticos onde os estados são representados por um vetor de características ou variáveis. Os valores dessas características ou variáveis mudam com base nas ações tomadas (no caso do RL) ou devido a eventos aleatórios (no caso do CSM). Um desafio neste contexto está em selecionar as variáveis de estado apropriadas e entender os eventos que podem ocorrer em cada estado. No entanto, no RL, o agente interage ativamente com o seu ambiente e aprende com essas experiências. Normalmente, um aprendizado eficaz no RL requer uma quantidade significativa de dados.

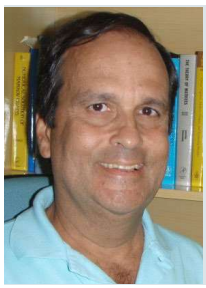
Há inúmeros outros exemplos dentro do campo do Aprendizado de Máquina baseado em modelos, incluindo modelos de Markov ocultos [5], Redes Bayesianas e Redes Neurais [6] (neste último caso, num sentido mais amplo onde são modeladas as relações entre as entradas e saídas). Devido a restrições de espaço, não podemos nos aprofundar nesses métodos neste texto, mas esperamos ter despertado a curiosidade do leitor para explorar esses modelos no futuro.

Existe uma grande sinergia entre a CSM e a ML baseado em modelos, ambos proporcionando um conjunto de ferramentas poderosas para compreender, prever e otimizar o desempenho de sistemas computacionais. Concluímos esse texto ressaltando a relevância de compreender o que é um modelo para o entendimento das tecnologias atuais e para a criação de novas tecnologias e algoritmos. Para lidar com esses modelos, é necessário possuir um nível básico de conhecimentos fundamentais, presentes no currículo das Ciências da

Computação ou nas Engenharias, porém estes têm sido por vezes negligenciados, substituindo-se pelo ensino sobre o uso de ferramentas e pacotes de software. É imprescindível compreender minimamente a teoria por detrás das "black-boxes" "caixas pretas" de IA, tanto para possibilitar o uso adequado dessas ferramentas e desenvolver uma visão crítica da tecnologia, quanto para avançar o estado da arte na área. A colaboração interdisciplinar também é crucial para promover a inovação e a utilização eficaz dos paradigmas de CSM e ML.

Referências:

1. A. G. Streit, G. H. A. Santos, R. M. M. Leão, E. de Souza e Silva, D. S. Menasché, and D. Towsley, "Network anomaly detection based on tensor decomposition," Comput. Networks, vol. 200, p. 108503, 2021.
2. F. Baskett, K. Chandy, R. Muntz, and F. Palacios, "Open, Closed and Mixed Networks of Queues with Different Classes of Customers," Journal of the ACM, vol. 22, pp. 248–260, 1975.
3. L. Kleinrock, Queueing Systems, Volume II: Computer Applications. Wiley-Interscience, 1976.
4. E. de Souza e Silva and H. R. Gail, "Calculating availability and performability measures of repairable computer systems using randomization," J. ACM, vol. 36, no. 1, pp. 171–193, 1989.
5. E. de Souza e Silva, R. M. M. Leão, and R. R. Muntz, "Performance Evaluation with Hidden Markov Models," ser. Lecture Notes in Computer Science, vol. 6821. Springer, 2010, pp. 112–128.
6. K. P. Murphy, Probabilistic Machine Learning: An Introduction. MIT Press, 2022.
7. T. M. Mitchell, "Machine Learning", McGraw-Hill Science, 1997.
8. Baseada na definição de Arthur Samuel (1959) https://en.wikipedia.org/wiki/Arthur_Samuel_(computer_scientist)



EDMUNDO DE SOUZA E SILVA é Professor Titular da COPPE/UFRJ. Obteve o doutorado em Ciência da Computação pela UCLA. Foi membro do CACC do CNPq e e, na CAPES, foi Coordenador de Área de Ciência da Computação na CAPES. Recebeu a comenda da Ordem Nacional de Mérito Científico em 2008. É membro da Academia Brasileira de Ciências e da Academia Nacional de Engenharia. Suas áreas de interesse incluem a modelagem e análise de sistemas de computação, redes de comunicação de dados e Aprendizado de Máquina.