



## Portfólio 3 - Redes Bayesianas

### 1 Introdução

Este projeto tem como principal objetivo apresentar a explicação do problema e o exemplo de uso do algoritmo desenvolvido utilizando redes Bayesianas. Este material é referente ao Portfólio 3 da disciplina de inteligência 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

O problema de falhas em um cluster de alto desempenho como o LNCC apresenta um grande obstáculo devido à incerteza e ao ruído causado pela gigantesca estrutura. O desafio fundamental é como modelar e tomar decisões racionais em sistemas complexos, onde as variáveis se influenciam mutuamente de maneiras sutis. No cluster, se manifesta na propagação causal, a tomada de decisões em sistemas complexos, onde falhas de infraestrutura, como falhas no hardware e erros na configuração do cluster, causam diretamente a falha de um nó específico, que, por sua vez, pode gerar múltiplos sintomas interligados como lentidão e KernelPanic(sistema operacional corrompido).

Portanto, o problema final é a necessidade de representar de forma compacta e eficiente as interdependências entre falhas e sintomas, permitindo descobrir as causas a partir dos efeitos observados.

### 3 Projeto utilizando conceito de redes bayesianas para solução de um problema

O código basicamente construiu uma Rede Bayesiana Discreta pra simular o caos do cluster.

Primeiro, veio o mapa do problema, que é o Grafo Dirigido Acíclico (DAG) sobre quem causa o quê. Por exemplo, definimos que falhas no hardware e erros na configuração são as causas raízes que apontam para falhas nos nós, e uma falha em um nó junto com sobrecarga na rede apontam para lentidão. Essa estrutura é a chave porque ela diz ao sistema quais variáveis só precisam se preocupar com seus "pais" imediatos.

Depois, preenchemos esse mapa com os números da incerteza isso é onde o conhecimento de domínio complexo de um Sysadmin é codificado. Por exemplo, a CPD de uma falha em um nó diz: "Se o Hardware e a Configuração estiverem OK, a chance do nó cair é minúscula (0.001); mas se ambos falharem, a chance é quase certa (0.99)". É o código dizendo o quão provável é um efeito dado uma combinação das suas causas. Essas probabilidades são fixas e representam a frequência ou o risco.

A cereja do bolo é a Inferência, feita pelo método usado para eliminar variáveis. Essa ferramenta faz a "mágica" reversa: em vez de ir da causa para o efeito, ela vai do efeito observado de volta para a causa mais provável.

No Cenário 1, por exemplo, ele viu o sintoma LogTimeout: 1 e usou isso para recalculer a probabilidade de ser uma sobrecarga na rede ou uma falha no hardware.

Ou seja, o código foi estruturado para decompor a complexidade, quantificar a incerteza com as CPDs, e, o mais importante, realizar o raciocínio probabilístico reverso para transformar logs e lentidão em um diagnóstico concreto de causa.

### 3.1 Exemplo de uso

```
Modelo de Cluster verificado com sucesso.

--- Cenário 1: Logs de Timeout Observados ---
Probabilidade de ser Rede: 0.2879
Probabilidade de ser Hardware: 0.0055

--- Cenário 2: Kernel Panic + Temperatura Alta ---
+-----+-----+-----+
| FalhaHardware | ErroConfig | phi(FalhaHardware,ErroConfig) |
+=====+=====+=====+
| FalhaHardware(0) | ErroConfig(0) | 0.0301 |
+-----+-----+-----+
| FalhaHardware(0) | ErroConfig(1) | 0.1026 |
+-----+-----+-----+
| FalhaHardware(1) | ErroConfig(0) | 0.8583 |
+-----+-----+-----+
| FalhaHardware(1) | ErroConfig(1) | 0.0090 |
+-----+-----+-----+

--- Cenário 3: Diagnóstico MAP (Timeout + Panic) ---
Finding Elimination Order: : 100%|
Eliminating: FalhaNo: 100%|
Causa mais provável: ErroConfig
```

Figura 1: Execução do algoritmo desenvolvido

## 4 Reflexão pessoal sobre o processo de desenvolvimento

A escolha desse problema caiu como uma luva pra mim. Como estou fazendo a matéria de Programação para Sistemas Paralelos e Distribuídos, escuto falar de clusters o tempo todo. Por isso, quando entendi como funcionam as Redes Bayesianas, a primeira coisa que me veio à mente foi aplicar essa lógica para tentar tratar os erros e fazer o diagnóstico de falhas dentro de um ambiente de configuração de um cluster de alto desempenho.

## 5 Desafios encontrados

Um grande desafio foi montar o modelo, pensando em como o problema ia ficar desenhado no mapa para o algoritmo de Rede Bayesiana poder trabalhar. A outra dor de cabeça foi acertar a mão em como uma falha interfere na outra dentro de um ambiente como o de um cluster. Tivemos que bolar as setas de quem causa o quê e depois botar os números que dizem o quão provável é um efeito quando a causa acontece.