

UNIVERSIDADE FEDERAL DE ALAGOAS INSTITUTO DA COMPUTAÇÃO INTELIGÊNCIA ARTIFICIAL 2025.1

PROFESSORES: EVANDRO COSTA E GLAUBER LEITE ALUNAS: FABIANA DE ALBUQUERQUE SILVA E LAÍS DAGNÉSIA

Lista 2

1) Desenvolva, usando a ferramenta Scikit Fuzzy (ou uma outra de sua livre escolha), um sistema baseado em regras fuzzy para Controle de Temperatura de um Chuveiro, contendo como referência a base de conhecimento a seguir, a qual, deverá ser melhorada, incluindo possível ampliação.

Sobre a solução a ser desenvolvida, apresente:

a) Descrição do processo de modelagem, mostrando e discutindo as funções de pertinência escolhidas e utilizadas para a etapa de fuzzificação, bem como a solução para etapa de defuzzificação), e discuta e mostre como é feita a inferência sobre o conhecimento, incluindo discussão sobre a etapa de agregação.

Resposta: A modelagem foi realizada utilizando a biblioteca Scikit-Fuzzy em Python. Modelagem do Sistema:

- Fuzzificação: As variáveis de entrada foram modeladas com funções de pertinência do tipo trapezoidal para os extremos e triangular para valores médios, devido à sua simplicidade e fácil interpretação.
 - Temperatura: baixa (0-30°C), média (20-40°C), alta (35-50°C).
 - Fluxo: baixo (0-5 l/min), médio (3-10 l/min), alto (8-15 l/min).
 - Abertura da Válvula: pequena (0-40%), moderada (30-70%), grande (60-100%).

Inferência: Foi aplicada a inferência Mamdani, utilizando o operador mínimo (AND) para ativação das regras e o operador máximo (OR) para agregação dos resultados de cada regra.

- Defuzzificação: A saída foi defuzzificada utilizando o método do centróide, que calcula o centro de massa da área agregada, fornecendo o valor crisp da abertura da válvula.

Base de Regras Fuzzy:

- 1. SE temperatura é baixa E fluxo de água é alto, ENTÃO abertura da válvula é grande.
- 2. SE temperatura é baixa E fluxo de água é médio, ENTÃO abertura da válvula é moderada.
- 3. SE temperatura é média E fluxo de água é alto, ENTÃO abertura da válvula é moderada.
- 4. SE temperatura é média E fluxo de água é baixo, ENTÃO abertura da válvula é pequena.
- 5. SE temperatura é alta E fluxo de água é baixo, ENTÃO abertura da válvula é pequena.
- 6. SE temperatura é alta E fluxo de água é alto, ENTÃO abertura da válvula é moderada.
- b) Execute e mostre esta aplicação rodando com o uso da ferramenta ScikitFuzzy, mostrada em aula. Como entrega relativa à questão, espera-se ainda uma documentação com todas as etapas citadas da modelagem, assim como:
- (i) Descrição do problema, indicando claramente os objetivos da aplicação, as variáveis usadas;

Resposta: O objetivo do sistema fuzzy é controlar a abertura da válvula de um chuveiro de forma a manter a temperatura desejada da água, mesmo com variações no fluxo. As variáveis de entrada são a temperatura da água e o fluxo de água, e a variável de saída é a abertura da válvula. O sistema deve ajustar a abertura da válvula conforme as regras estabelecidas, garantindo conforto e economia.

(ii) Descrição da solução;

Resposta: A solução proposta utiliza Lógica Fuzzy com inferência do tipo Mamdani para modelar o sistema de controle do chuveiro. As etapas principais são:

- Fuzzificação das variáveis de entrada (Temperatura e Fluxo de Água) utilizando funções de pertinência trapezoidais e triangulares.
- Definição de regras fuzzy baseadas no conhecimento do problema.
- Inferência Mamdani, com agregação pelo operador máximo (OR).

- Defuzzificação utilizando o método do centróide para determinar o valor crisp da abertura da válvula.
- (iii) Compartilhe o notebook utilizado pelo uso do Scikit-Fuzzy

Link do notebook no Colab: Controle Fuzzy do Chuveiro (Scikit-Fuzzy)

2. Desenvolva, usando a ferramenta Scikit Fuzzy um sistema baseado em regras fuzzy para análise de risco em projetos de desenvolvimento de software, onde a base de regras deverá ser obtida automaticamente a partir de uma base de dados relacionada ao problema, usando aprendizado de máquina supervisionado para gerar uma árvore de decisão (fuzzy) ou então regras (fuzzy) diretamente.

A solução a ser desenvolvida deve ser descrita e apresentada:

2.1 Construção e modelagem da base de regras.

Resposta: Para construir a base de regras fuzzy, seguimos os seguintes passos:

Base de dados: Como não foi fornecida uma base real, criamos uma base sintética com 1000 amostras simulando projetos de software. As variáveis usadas foram:

- Experiência da equipe (0 a 10): quanto maior, mais experiente.
- Tempo estimado (1 a 52 semanas): tempo planejado para concluir o projeto.
- Orçamento (10 a 500 mil): recursos financeiros disponíveis.
- Complexidade (0 a 10): quão complexo é o projeto.
- Risco (0 a 10): calculado por uma fórmula heurística que considera menor experiência, maior complexidade e tempo, e menor orçamento como fatores de maior risco.

Depois disso, transformamos o risco contínuo em categorias discretas:

0: Risco baixo

• 1: Risco médio

• 2: Risco alto

Geração automática de regras: Utilizamos uma árvore de decisão (DecisionTreeClassifier) para identificar padrões lógicos entre as variáveis de entrada e a categoria de risco. Cada caminho na árvore representa uma regra de decisão, e essas regras foram convertidas para o sistema fuzzy. Exemplo de regra derivada da árvore: "Se a experiência é baixa e a complexidade é alta, então o risco é alto."

2.2 Descrição do processo de modelagem, mostrando e discutindo as funções de pertinência escolhidas e utilizadas para a etapa de fuzzificação, bem como a solução para etapa de defuzzificação), e discuta e mostre como é feita a inferência sobre o conhecimento, incluindo discussão sobre a etapa de agregação.

Resposta: Funções de pertinência (fuzzificação): Para transformar variáveis numéricas em valores linguísticos fuzzy, usamos a função automf(3) da biblioteca Scikit-Fuzzy, que cria automaticamente três categorias fuzzy para cada variável:

• Baixo (poor);

• Médio (average);

• Alto (good).

Inferência (Mamdani)

Usamos a metodologia Mamdani, amplamente utilizada em sistemas fuzzy:

Etapas:

1. Fuzzificação: convertemos as entradas numéricas em graus de pertinência.

2. Avaliação das regras: cada regra fuzzy é ativada com base nas entradas.

3. Agregação: as saídas das regras são combinadas com o operador max.

4. Defuzzificação: usamos o método do centróide (centroid) para transformar a saída fuzzy em um valor numérico final.

Regras fuzzy aplicadas

A partir da árvore, criamos as seguintes regras (exemplares):

- Regra 1: Se experiência é baixa e complexidade é alta, então o risco é alto.
- Regra 2: Se experiência é média e orçamento é médio, então o risco é médio.
- Regra 3: Se experiência é alta e complexidade é baixa, então o risco é baixo.
- 2.3 Execute e mostre esta aplicação rodando com o uso da ferramenta ScikitFuzzy, mostrada em aula. Como entrega relativa à questão, espera-se ainda uma documentação com todas as etapas citadas da modelagem, assim como:
- (i) Descrição do problema, indicando claramente os objetivos da aplicação, as variáveis usadas;

Resposta: O objetivo é construir um sistema fuzzy para análise de risco em projetos de software, utilizando variáveis como experiência, tempo, orçamento e complexidade. A meta é gerar regras automaticamente a partir de dados reais ou sintéticos, simulando um cenário de tomada de decisão.

(ii) Descrição da solução;

Resposta:

- Uma base sintética foi criada para simular os dados do problema.
- Usou-se aprendizado de máquina (árvore de decisão) para gerar regras fuzzy automaticamente.
- Utilizou-se Scikit-Fuzzy para construir o sistema fuzzy Mamdani.

- Foram definidas funções de pertinência e regras baseadas na árvore.
- A inferência fuzzy foi realizada e o valor de risco foi obtido pela defuzzificação.

(iii) Compartilhe o notebook utilizado pelo uso do Scikit-Fuzzy

Link do notebook: Sistema Fuzzy Risco Projeto.ipynb

3. Desenvolva, discuta e apresente um exemplo de aplicação de rede bayesiana para realizar um sistema de diagnóstico médico, explicando a modelagem das relações entre os sintomas e o diagnóstico da doença. Apresente uma solução de representação de conhecimento que você construiu e mostre o funcionamento da inferência. Utilize a ferramenta nética (ou alguma outra de sua livre escolha) para realizar essa aplicação.

(i) Descrição da Aplicação

O desenvolvimento é de um sistema de diagnóstico médico probabilístico com base em rede bayesiana, com o objetivo de estimar o risco de infarto do miocárdio (IAM). O sistema considera dados clínicos, laboratoriais, histórico familiar, sintomas e especialmente variáveis de ressonância magnética cardíaca (RMC) do ventrículo esquerdo. O diagnóstico final não é apenas "sim/não", mas classificado em níveis de risco: baixo, moderado e alto, com probabilidades associadas.

(ii) Modelagem das Relações entre Sintomas e Diagnóstico

A rede bayesiana segue a estrutura:

> Fatores de risco:

Hipertensão, diabetes, IMC, sedentarismo, aneurisma etc.

> Exames específicos:

LDL, escore de cálcio, ITB, TFG, retinopatia etc.

> Histórico familiar:

Idade de risco (homens ≥ 45 , mulheres ≥ 55), histórico de diabetes, parentes com DAC.

> Sintomas:

Dor no peito, fadiga, náusea, falta de ar, tontura.

> RMC – Variáveis prioritárias:

Realce tardio (LGE), Fração de ejeção (FEVE), Alterações na contratilidade regional e Espessamento anormal do ventrículo esquerdo.

> Saída:

Risco de Infarto: baixo, moderado, alto.

(iii) Representação do Conhecimento

A rede foi implementada com a biblioteca pgmpy (Python).

(iv) Funcionamento da Inferência

Ao fornecer evidências clínicas e de RMC, o sistema calcula a distribuição de probabilidade dos níveis de risco de infarto.

Exemplo de saída:

Infarto	Probability
Baixo	0.04
Moderado	0.28
Alto	0.68

(v) Discussão com Base no Sistema de Alarme (Russell & Norvig)

Assim como o exemplo clássico do livro, onde o alarme depende de eventos (terremoto ou roubo) e causa chamadas dos vizinhos, aqui temos:

- Sintomas e exames como evidências observáveis;
- Fatores de risco como causas latentes;

• Infarto como variável oculta a ser inferida;

• A probabilidade de infarto é ajustada dado o contexto clínico e imagem.

Este sistema bayesiano combina conhecimento médico com uma estrutura probabilística

formal para ajudar no diagnóstico de infarto do miocárdio, sendo capaz de incorporar

evidências clínicas e de imagem (RMC) com precisão. Além de ser explicável, é extensível

para bases reais e integração com IA baseada em imagem no futuro.

4. Desenvolva um sistema de gerenciamento de riscos em projetos de software baseado em

Redes Bayesianas.

Resposta:

Link do código do sistema: o questao4.ipynb

5. Sobre a técnica de raciocínio baseado em casos, desenvolva um protótipo funcional de um

sistema de diagnóstico médico baseado em casos para identificar possíveis doenças com base

em sintomas relatados pelo paciente. Os sintomas fornecidos serão comparados com casos

anteriores para identificar possíveis doenças e recomendar tratamentos. Apresente solução

para as seguintes etapas:

(5.1) Indexação e representação de casos: crie uma base de casos contendo informações sobre

diagnósticos médicos anteriores, incluindo sintomas relatados pelo paciente e doença

diagnosticada.

(5.2) Recuperação de casos: Implemente uma função que permita ao Sistema receber como

entrada sintomas relatados pelo paciente, daí utilize um algoritmo de recuperação de casos

para encontrar casos similares na base de casos, comparando os sintomas relatados pelo

paciente com os casos armazenados.

(5.3) Apresente o diagnóstico.

(5.4) Use uma ferramenta para realizar essa aplicação (ex.: MyCBR ou JColibri ou

Scikit-CBR ou PyCBR)

Resposta:

Link do código: https://github.com/laisdagnesia/IA UFAL/blob/main/questao5.py