Lógica

Prof. Dr. Rafael Teixeira Sousa

UFMT

Outline

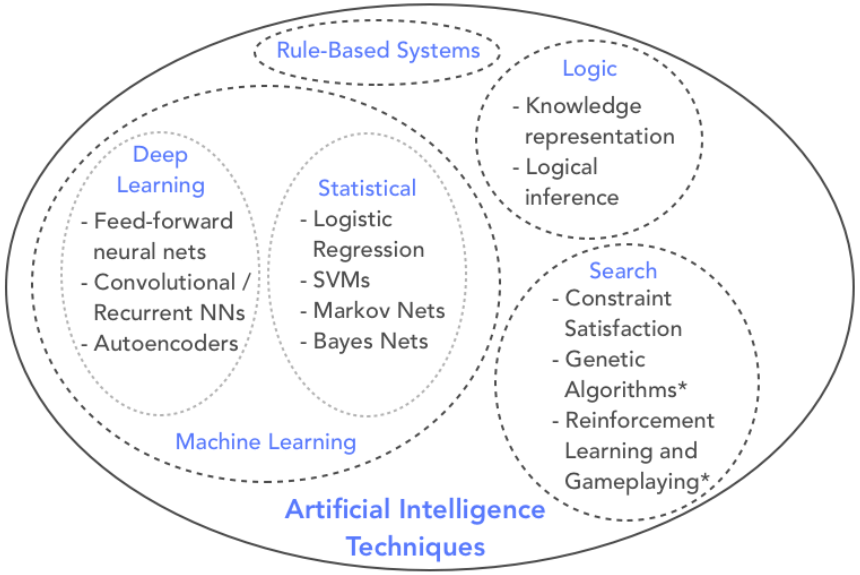
- Lógica proposicional
- Inferência Lógica
- Lógica de primeira ordem
- Lógica Fuzzy

Questão

- Se $X_1 + X_2 = 10$ e $X_1 - X_2 = 4$, qual é o valor de X_1?

Contexto histórico

- Lógica e Busca foram dominantes em IA até 90
  - Vantagem:
    - Expressividade
  - Problemas:
    - Determinística. Não lida com incertezas
    - Baseada em regras. Não permite ajuste com base em dados



Linguagem Natural

- Exemplo 1:
  - Um **real** é melhor que um **centavo**
  - Um **centavo** é melhor do que **nada**
  - Logo, um **real** é melhor do que **nada**

- Exemplo 2:
  - Um **real** é melhor do que **nada**
  - **Nada** é melhor do que a **paz mundial**
  - Logo, um **real** é melhor do que a **paz mundial**?

Linguagem

- Uma linguagem é um mecanismo de expressão
- Linguagem Natural:
  - Português: Dois divide números pares
  - Inglês: Two divides even numbers
- Linguagem de programação:
  - Python: $def\ even(x)$: $return\ x$ % 2 == 0
  - C++: bool even(int x) { return x % 2 == 0; }
- Linguagem lógica:
  - Lógica de primera ordem: $\forall x.\, \mathrm{Even}(x) \rightarrow \mathrm{Divides}(x, 2)$

Objetivos da linguagem lógica

- Representar conhecimento sobre o mundo

- Raciocinar com o conhecimento

Exercício

- Qual dos argumentos é valido?
  1. Se neva, então faz frio. Está nevando. Logo, está fazendo frio.
  2. Se chove, então a rua fica molhada. A rua está molhada. Logo, choveu.

Exercício

- Qual dos argumentos é valido?
  1. Se neva, então faz frio. Está nevando. Logo, está fazendo frio.
  2. Se chove, então a rua fica molhada. A rua está molhada. Logo, choveu.

1. Nevar → Frio
   - Se **Nevar** é verdadeiro, logo, está **Frio**

2. Chove → Molhado

Ingredientes de uma lógica: Sintaxe

- Define um conjunto válido de fórmulas
- Símbolos:
  - Proposições: $a, b, c, \ldots$
  - Conectivos: $\neg, \wedge, \vee, \rightarrow$
- Fórmulas (f):
  - Se α e β são fórmulas, então:
    - $\neg\, \alpha$
    - $\alpha \wedge \beta$
    - $\alpha \vee \beta$
    - $\alpha \rightarrow \beta$

Ingredientes de uma lógica: Semântica

- Define o significado das sentenças
- Interpretação: Associação entre proposições e valores-verdade (V ou F)

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \rightarrow q$
F	F	V	F	F	V
F	V	V	F	V	V
V	F	F	F	V	F
V	V	F	V	V	V

- Tipos de fórmula:
  - **Válida** (tautológica): é verdadeira em **toda** interpretação
  - **Satisfatível** (contingente): é verdadeira em **alguma** interpretação
  - **Insatisfatível** (contraditória): é verdadeira em **nenhuma** interpretação

Sintaxe vs Semântica

- Sintaxe: o que são expressões válidas?
- Semântica: o que as expressões significam?

- Diferente sintaxe, mesma semântica:
  - $2 + 3 \Leftrightarrow 3 + 2$
- Mesma sintaxe, diferente semânticas (1 vs 1,5):
  - 3/2 (Python 2.7) $\not\Leftrightarrow$ 3/2 (Python 3)

Lógicas

- **Proposicional**
- Primeira ordem
- Segunda ordem

Exemplo – Lógica Proposicional

- Está chovendo
- **Se** está chovendo, **então** a rua está molhada
- **Se** a rua está molhada, **então** a rua está escorregadia

- Vocabulário
  - c: “está chovendo”
  - m: “a rua está molhada”
  - e: “a rua está escorregadia”
- Formalização
  - $KB = \{c, c \rightarrow m, m \rightarrow e\}$

Base de conhecimento (KB – Knowledge Base)

- Conjunto de fórmulas
  - $\Delta = \{c, c \rightarrow m, m \rightarrow e\}$

- Aumentar as fórmulas aumentas as restrições
- Criamos novas fórmulas a partir do conhecimento prévio e de regras de inferência lógica

Inferência lógica - Prova por dedução

- Exemplo:
  - Está chovendo (c)
  - **Se** está chovendo, **então** a rua está molhada ($c \rightarrow m$)
  - Logo, está molhado (m)

$$\frac{c,\ c \rightarrow m}{m} \qquad \frac{(premissa)}{(conclusão)}$$

  - Regra de inferência: Modus ponens
    - Para qualquer símbolos p e q:

$$\frac{p,\ p \rightarrow q}{q}$$

Inferência lógica - Prova por dedução

- Uma regra de inferência é definida como:
  - Se $f_1, \ldots, f_k, g$ são fórmulas, então:

$$\frac{f_1, \ldots, f_k}{g}$$

- Se as premissas são verdadeiras (estão na KB), então a conclusão também é verdadeira e pode ser incluída na KB.

Algoritmo de inferência

- Input: Um conjunto de regras (fórmulas)
- Repita até não haver mudança na KB:
  - Escolha um conjunto de fórmulas $f_1, \ldots, f_k \in KB$
  - Se existe uma regra $\frac{f_1,\ldots,f_k}{g}$:
    - Adicione g a KB
- KB prova/deriva f ($\text{KB} \vdash f$) se, e somente se, f for adicionada a KB

Exemplo

- Vocabulário
  - c: “está chovendo”
  - m: “a rua está molhada”
  - e: “a rua está escorregadia”
- Ponto inicial:
  - $KB = \{c, c \rightarrow m, m \rightarrow e\}$
- Aplicando modus ponens para c e $c \rightarrow m$:
  - $KB = \{c, c \rightarrow m, m \rightarrow e, m\}$
- Aplicando modus ponens para m e m → e:
  - $KB = \{c, c \rightarrow m, m \rightarrow e, m, e\}$
- Não podemos derivar fórmulas como: $\neg m$ e $c \rightarrow e$

Corretude e completude

"Juro pela minha honra dizer toda a verdade e só a verdade"

- Corretude: "só a verdade"
  - Que todas as inferências sejam com base nas **premissas verdadeiras** e conforme a semântica e sintaxe
- Completude: "toda a verdade"
  - Que sejam exploradas **todas as inferências**

Corretude: Exemplo

$$\frac{c, \quad c \rightarrow m}{m}$$

- Correto?

Corretude

$$\frac{c, \quad c \to m}{m}$$

- Correto? Sim

$$\frac{m, \quad c \to m}{c}$$

- Correto?

Corretude

$$\frac{c, \quad c \rightarrow m}{m}$$

- Correto? Sim

$$\frac{m, \quad c \rightarrow m}{c}$$

- Correto? Não

Completude

- Vocabulário
  - c: “está chovendo”
  - m: “a rua está molhada”
  - n: “está nevando”
- Formalização
  - $KB = \{c, c \vee n \rightarrow m\}$
  - $f = m$
  - Regras: $\left\{\frac{f, f \rightarrow g}{g}\right\}$ (Modus ponens)
- Semanticamente m é verdadeiro, mas sintaticamente não, pois não possuímos uma regra de inferência capaz de provar m
- Modus ponens é **incompleto**

Corrigindo Completude

- Alternativa 1:
  - Restringir as fórmulas
  - Lógica proposicional ⇒ Lógica proposicional com apenas clausulas de Horn
- Alternativa 2:
  - Usar outras regras de inferência

Rules of Inference (Table 1.3.1- Page 39)

Modus Ponens $p \rightarrow q$ p $\therefore q$	Modus Tollens $p \rightarrow q$ $\sim q$ $\therefore \sim p$	Disjunctive Syllogism $p \vee q$ | $p \vee q$ $\sim q$ | $\sim p$ $\therefore p$ | $\therefore q$
Disjunctive Addition p | q $\therefore p \vee q$ | $\therefore p \vee q$	Conjunctive Simplification $p \wedge q$ | $p \wedge q$ $\therefore p$ | $\therefore q$	Rule of Contradiction $\sim p \rightarrow c$ $\therefore p$
Hypothetical Syllogism $p \rightarrow q$ $q \rightarrow r$ $\therefore p \rightarrow r$	Conjunctive Addition p q $\therefore p \wedge q$	Dilemma $p \vee q$ $p \rightarrow r$ $q \rightarrow r$ $\therefore r$

Cláusulas de Horn

- Cláusulas apenas nos formatos:

$$(p_1 \wedge \cdots \wedge p_k \rightarrow q)$$

$$(p_1 \wedge \cdots \wedge p_k \rightarrow falso)$$

Cláusulas de Horn

- Cláusulas apenas nos formatos:

$$(p_1 \wedge \cdots \wedge p_k \rightarrow q)$$

$$(p_1 \wedge \cdots \wedge p_k \rightarrow falso)$$

Modus ponens com cláusulas de Horn pode ser definido como:

$$\frac{(p_1 \wedge \cdots \wedge p_k \rightarrow q) \rightarrow q}{q}$$

Supondo que a KB contenha somente cláusulas de Horn, então todas as fórmulas podem ser derivadas com o uso consecutivo de Modus ponens

Limitações da lógica proposicional

- Alice e Enzo sabem matemática
  - AliceSabeMatemática ∧ EnzoSabeMatemática
- Todos estudantes sabem matemática
  - AliceéEstudante → AliceSabeMatemática
  - EnzoéEstudante → EnzoSabeMatemática
  - ...
- Todo inteiro par maior que 2 é a soma de dois primos
  - ???

Limitações da lógica proposicional

- Alice e Enzo sabem matemática
  - **AliceSabeMatemática** ∧ **EnzoSabeMatemática**
- Todos estudantes sabem matemática
  - **AliceéEstudante** → **AliceSabeMatemática**
  - **EnzoéEstudante** → **EnzoSabeMatemática**
  - ...
- Todo inteiro par maior que 2 é a soma de dois primos
  - ???
- Falta:
  - **Objetos e predicados**: **AliceSabeMatemática** possui estrutura interna (**Alice**, **Sabe**, **Matemática**)
  - **Quantificadores e variáveis**: "Todos" é um quantificador

Exemplo

sist_especialista.py

Lógica de Primeira Ordem

- Alice e Enzo sabem matemática

$$Sabe(alice, Matemática) \wedge Sabe(enzo, Matemática)$$

- Todos estudantes sabem matemática

$$\forall x\ Estudante(x) \rightarrow Sabe(x, Matemática)$$

Sintaxe

- Termos:
  - Constantes ($Matemática$)
  - Variáveis (x)
  - Funções ($Soma(3, x)$)
- Fórmulas:
  - Fórmula atómica ($Sabe(x, Matemática)$)
  - Conectivo ($Estudante(x) \rightarrow Sabe(x, Matemática)$)
  - **Quantificador** ($\forall x\ Estudante(x) \rightarrow Sabe(x, Matemática)$)

Quantificadores

- Quantificador universal (∀)
  - Conjunção: $\forall x\ P(x)$ é como $P(A) \wedge P(B) \wedge \cdots$
- Quantificador existencial (∃)
  - Disjunção: $\exists x\ P(x)$ é como $P(A) \vee P(B) \vee \cdots$

- Propriedades:
  - $\neg \forall x P(x)$ equivalente a $\exists x \neg P(x)$
  - $\forall x \exists y\ Sabe(x, y)$ bem diferente de $\exists y \forall x\ Sabe(x, y)$

Quantificadores em linguagem natural

- Todo estudante sabe matemática
  - $\forall x\ Estudante(x) \rightarrow Sabe(x, Matemática)$

- Alguns estudantes sabem matemática
  - $\exists x\ Estudante(x) \wedge Sabe(x, Matemática)$

- Todo inteiro par maior que 2 é a soma de dois primos
  - ?

Quantificadores em linguagem natural

- Todo estudante sabe matemática
  - $\forall x\ Estudante(x) \rightarrow Sabe(x, Matemática)$

- Alguns estudantes sabem matemática
  - $\exists x\ Estudante(x) \wedge Sabe(x, Matemática)$

- Todo inteiro par maior que 2 é a soma de dois primos
  - $\forall x\ InteiroPar(x) \wedge MaiorQue(x, 2) \rightarrow \exists y \exists z\ Igual(x, Soma(y, z)) \wedge Primo(y) \wedge Primo(z)$

Resumo

- Lógica
  - Representar conhecimento e relações para inferir conclusões lógicas
  - Primeira abordagem de IA a ter uso prático
- Lógica Proposicional
  - Abordagem simples e direta
  - Pode ser completa com restrição a cláusulas de Horn
  - Difícil representação
- Lógica de primeira ordem
  - Introduz sintaxe aprimorada
- Lógica de segunda ordem
  - Introduz conjuntos
- Ordem superior

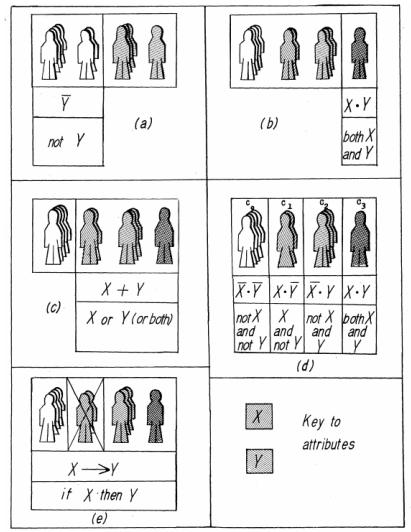


Fig. 1. Combinations of attributes.

LISP

- John McCarthy – 1958
- Família de linguagens voltadas a expressões matemáticas e símbolos

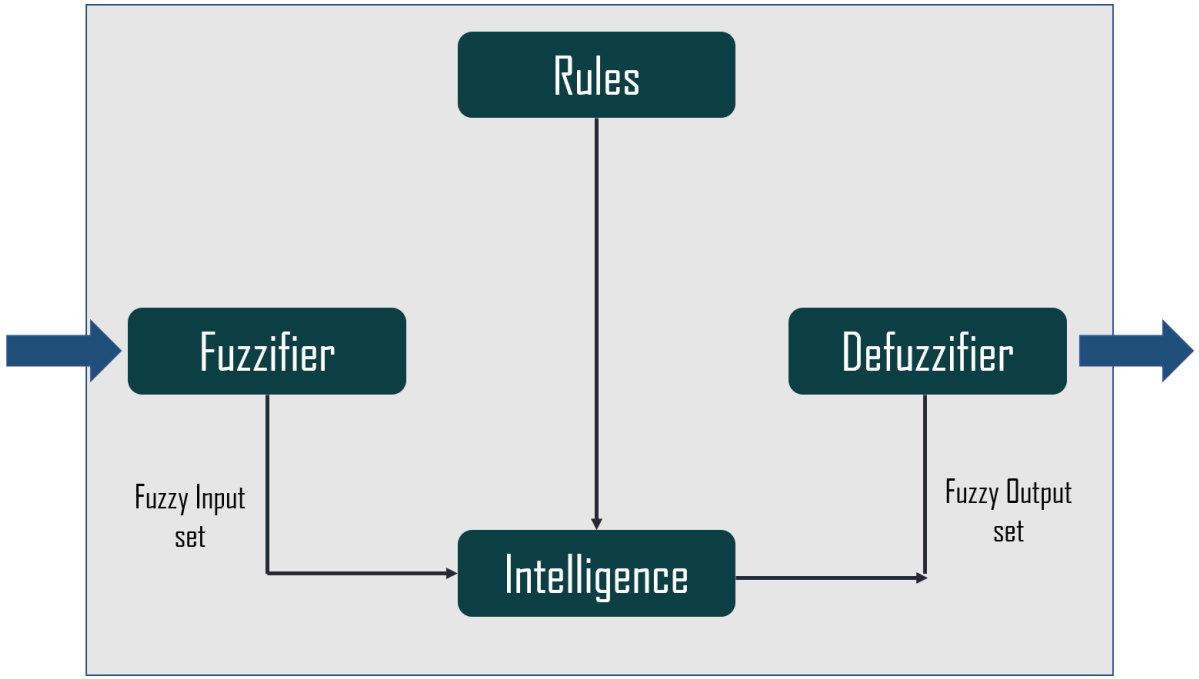
Lógica Fuzzy

- Exemplo:
  - Homens de meia idade
  - Lógica clássica
    - Se $40 \leq idade \leq 55 \rightarrow$ Homem de meia idade
  - Lógica Fuzzy

Idade	35	40	45	50	55
Grau de pertinência	0,0	0,5	1,0	0,5	0,0

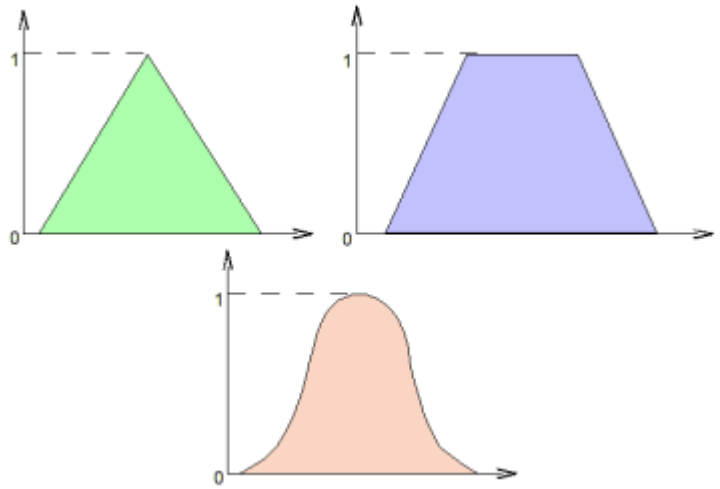
  - Funções de pertinência
    - Valor $\mu_x \epsilon [0,1]$

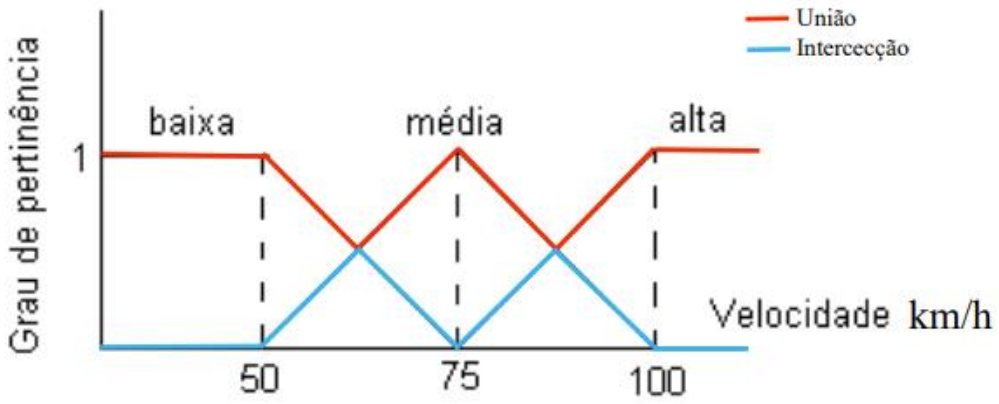
Fuzzificação



Funções de pertinência

- Triangular
- Gaussiana
- Trapezoidal





Aplicações

- Sistemas especialistas
- Demonstrações matemáticas
- Detecção de fraudes

- Exemplo:
  ar_fuzzy.py