Algorithmen, KI und Data Science 1 (AKIDS 1): **Expert Systems & Numerical Optimization**

Herbst- / Wintersemester 2023/24

Prof. Dr. Goran Glavaš

Benedikt Ebing

Fabian David Schmidt

Expert Systems

Exercise 1.1

Explain the terms symbolism and connectionism?
What are the differences?

Exercise 1.1 – Solution

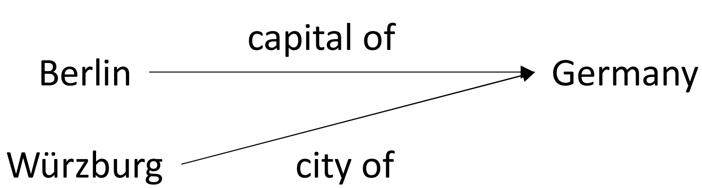
- Symbolic AI:
  - discrete, human interpretable
  - Knowledge represented with symbols
  - Inference: Formal symbolic (rule-based) reasoning over KB

- Connectionist AI:
  - Continous, mostly not human interpretable
  - Knowledge: Learned from raw data
  - Inference: Computations in a continous representation space

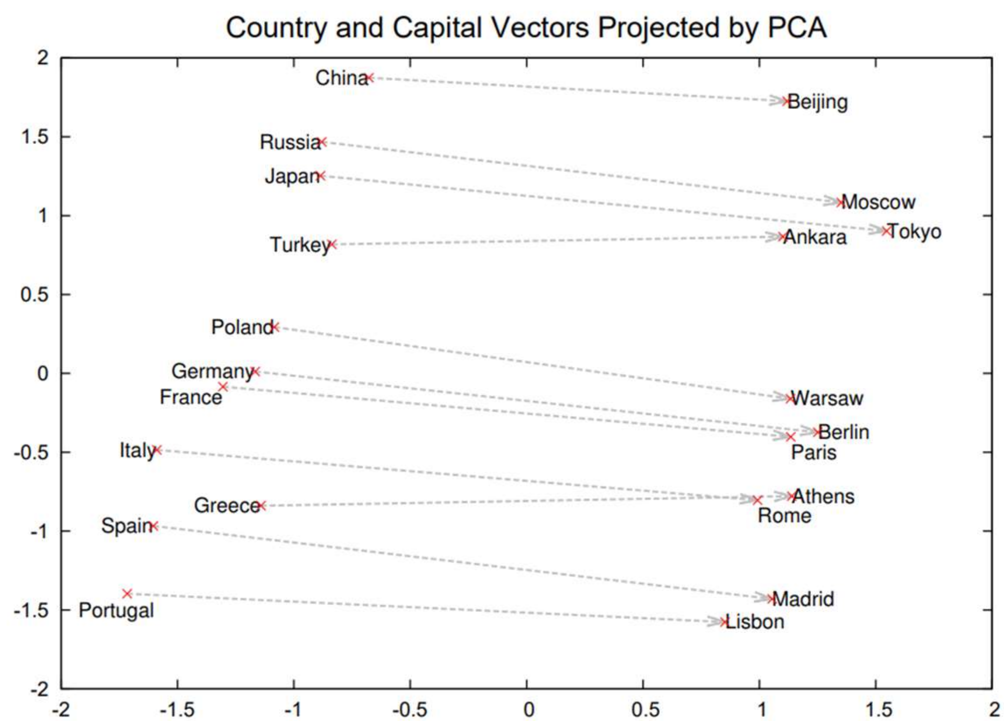
- Differences:
  - Discrete vs. continuous
  - Human interpretable vs. not human interpretable

Exercise 1.1 – Example

Symbolic AI



Connectionist AI



[Mikolov, Tomas, et al. (2013).]

Exercise 1.2

Recap: Propositional logic

Recap: Propositional logic

- Propostional variables $V = \{A, B, C, \dots\}$
- Logical operators
  - Negation ($\neg$)
  - Disjunction (OR, $\vee$)
  - Conjunction (AND, $\wedge$)
  - Implication ($\rightarrow$)
  - Equivalence ($\leftrightarrow$)
- Logical constants (True $\top$, False $\bot$)
- Parentheses

Recap: Propositional Logic

F	G	$\neg F$	$F \wedge G$	$F \vee G$	$F \rightarrow G$	$F \leftrightarrow G$
$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\top$	$\top$
$\bot$	$\top$	$\top$	$\bot$	$\top$	$\top$	$\bot$
$\top$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$
$\top$	$\top$	$\bot$	$\top$	$\top$	$\top$	$\top$

According to some political pundits, a person who is radical (R) is electable (E) if he/she is conservative (C), but otherwise is not electable. Which of the following is correct? Explain.

Exercise 1.2 – Solution

- $(R \land E\ ) \leftrightarrow C$
  - No, states that all conservatives are radical which is not what is stated in the text
- $R \rightarrow (E \leftrightarrow C)$
  - Correct, a radical person is electable if he/she is conservative.
- $R \rightarrow \big((C \rightarrow E) \lor \neg E\big)$
  - No, states that all radicals are electable

Exercise 1.3

Which of the following logical consequences are correct?

Hint: Formula G is a logical consequence of formula F ($F \Longrightarrow G$) if and only if every assignment that satisfies F also satisfies G.

Exercise 1.3 – Solution

- $False \rightarrow \text{True}$
  - Correct
- $True \rightarrow \text{False}$
  - Incorrect
- $(A \land B) \rightarrow (A \leftrightarrow B)$
  - Correct, left-hand side (lhs) has only one assignment that satisfies, this assignment is satisfied on the right-hand side (rhs) as well
- $(A \leftrightarrow B) \rightarrow (A \lor B)$
  - Incorrect, lhs is satisfied for $A = False$ and $B = False$, rhs not

Exercise 1.3 – Solution

- $(A \leftrightarrow B) \rightarrow (\neg A \vee B)$
  - Correct, rhs equivalent to $(A \rightarrow B)$
- $((A \wedge B) \rightarrow C) \rightarrow ((A \rightarrow C) \vee (B \rightarrow C))$
  - Correct, rhs is only $False$ for $A = True, B = True$ and $C = False$, in that case is lhs $False$ as well
- $(A \vee B) \wedge (\neg C \vee \neg D \vee E) \rightarrow (A \vee B)$
  - Correct, removing a conjunction on the rhs only allows for more assignments
- $(A \vee B) \wedge (\neg C \vee \neg D \vee E) \rightarrow (A \vee B) \wedge (\neg D \vee E)$
  - Incorrect, removing a disjunction on the rhs allows for less assignments to satisfy on the rhs

Exercise 1.4

Recap: Chaining

Recap: Forward vs. Backward chaining

- Forward chaining:
  - Starting from known data and advancing towards a conclusion
  - **To use:** when there is a small amount of data and a large space of possible solutions

- Backward chaining:
  - Choosing a possible conclusion (hypothesis) and trying to prove that it is valid by finding evidence
  - **To use:** Not too many possible conclusions, the amount of known data is large

Recap: Expert System Example

Shape: elongated | circular | rounded
Surface: smooth | coarse
Color: green | yellow | brown-yellow |
red | blue | orange
No. seeds: 0 | 1 | >1

R_1: **IF Shape** = elongated & **Color** = green | yellow **THEN Fruit =** banana

R_2: **IF Shape** = circular | rounded & **Diameter** = >10cm **THEN Fruit Type =** vine

R_3: **IF Shape** = circular & **Diameter** = <10cm **THEN Fruit Type =** tree

R_4: **IF No. Seeds** = 1 **THEN Seed Type =** bony

Recap: Backward Chaining Steps

1. Put goal variable onto (empty) stack
2. Find all rules with variable from the stack top on RHS
   1. If no rule has the stack-top variable on the RHS ➔ Ask the user
3. For each such rule:
   1. If LHS satisfied (all variables have correct values in WM)
      - Apply the rule (place the RHS variable and value in WM)
      - Remove the current goal from the stack
      - Continue form Step 2
   2. IF LHS not satisfied and value of some variable different to WM
      - Do not apply rule
   3. If LHS not satisfied and value of some variable missing in WM
      - Add variable to stack
      - Continue from Step 2

Exercise 1.4 - Solution

Step	Stack	WM	Conflicting Rules	Action
0	Fruit		R1, R6, R7, R8, R9, R10, R11, R12, R13, **R14**	Add Fruit Type to stack
1	Fruit Type Fruit		R2, **R3**	Ask user for Shape and add to WM
2	Fruit Type Fruit	Shape: circular	R2, **R3**	Ask user for Diameter and add to WM
3	Fruit Type Fruit	Shape: circular, Diameter: <10cm	R2, **R3**	Add fruit type to WM and pop from stack
4	Fruit	Shape: circular, Diameter: <10cm, Fruit Type: Tree	R1, R6, R7, R8, R9, R10, R11, R12, R13, **R14**	Ask user for No. Seeds and add to WM

Exercise 1.4 - Solution

Step	Stack	WM	Conflicting Rules	Action
5	Seed Type Fruit	Shape: circular, Diameter: <10cm, Fruit Type: Tree, Color: green	R1, R6, R7, R8, R9, R10, R11, R12, R13, **R14**	Add Seed Type to Stack
6	Seed Type Fruit	Shape: circular, Diameter: <10cm, Fruit Type: Tree, Color: green	R4, **R5**	As user for No. Seeds and add to WM
7	Seed Type Fruit	Shape: circular, Diameter: <10cm, Fruit Type: Tree, Color: green, No. Seeds: >1	R4, **R5**	Add Seed Type to WM and pop from Stack
8	Fruit	Shape: circular, Diameter: <10cm, Fruit Type: Tree, Color: green, No. Seeds: >1, Seed Type: multiple	R1, R6, R7, R8, R9, R10, R11, R12, **R13**, R14	Add Fruit=Apple to WM and pop stack -> Done

Exercise 1.5

Recap: Backward Chaining Algorithm

```
backward_chain(ont, rules, goal)
  s = [] # empty stack
  s.push(goal)
  wm = {} # empty hash table
  while not s.is_empty()
    goal = s.peek()
    matches = find_rules(rules, goal)
    if len(matches) == 0 # no rule with stack-top varia
      val = ask_user(goal)
      if value_valid(ont, val, goal)
        wm[goal] = val
      else
        return „error"
    for m in matches
      status = rule_status(m, wm)
      if status == True # LHS satisfied
        apply_rule(m, wm) # RHS added to wm
        s.pop()
        break
      elif status == False # LHS in conflict with wm
        continue
      else # status is a variable not in wm
        s.push(status)
        break
  return wm[goal]
```

```
value_valid(ont, var, val)
   vals = ont[var]
   if val in vals # hashtable lookup
     return True
   else
     return False

rule_status(rule, wm)
  for var in rule.LHS
    if var not in wm
      return var # not in wm
    elif rule.LHS[var] ≠ wm[var]
      return False # in wm, wrong val
  return True

apply_rule(rule, wm)
  var = rule.RHS.var
  val = rule.RHS.val
  wm[var] = val

find_rules(rules, goal)
  matches = []
  for rule in rules
    if rule.RHS.var == goal
      matches.append(rule)
  return matches
```

Numerical Optimization

Exercise 2.1 - Solution

- Newton´s Method:
  - $x^{k+1} = x^k - \frac{f(x^k)}{f'(x^k)}$
  - $x^{k+1} = x^k - \frac{\mathrm{f}'(\mathrm{x}^\mathrm{k})}{f''(x^k)}$
- Looking for $f'(x) = 0$
- Given: $f(x) = 3x^3 + 6x^2 + x - 3; x^0 = 1$
- Derivatives:
  - $f'(x) = 9x^2 + 12x + 1$
  - $f''(x) = 18x + 12$

Exercise 2.1 - Solution

- $x^{k+1} = x^k - \frac{f'(x^k)}{f''(x^k)}$
- $f'(x) = 9x^2 + 12x + 1$
- $f''(x) = 18x + 12$

- *Computation:*
  - $x^1 = 1 - \frac{9\times1^2+12\times1+1}{18\times1+1} = 0.2667$
  - $x^2 = -0.0214; x^3 = -0.0857; \boldsymbol{x^4 = -0.0893}$
  - Starting in $x^0 = -2$ leads to convergence at maximum $x = -1.244$

Exercise 2.1 - Solution

- Gradient Descent:
  - $x^{k+1} = x^k - \eta f'(x^k)$
- Looking for $f'(x) = 0$
- Given: $f(x) = 3x^3 + 6x^2 + x - 3; x^0 = 1$
- Derivatives:
  - $f'(x) = 9x^2 + 12x + 1$
- *Computation:*
  - $x^1 = 1 \ - 0.1\,(9 \times 1^2 + 12 \times 1 + 1) = 1.2$
  - $x^2 = -1.156; x^3 = -1.07; \boldsymbol{x^4 = 0.91}$
  - Starting in $x^0 = -2$ leads to divergence.