**Systemy sztucznej inteligencji**

**PROJEKT 2**

**Łukasz Dutka**

**Paweł Moniewski**

**Informatyka sem. V**

**grupa B**

**30.01.2016**

1. **Problematyka zadania**

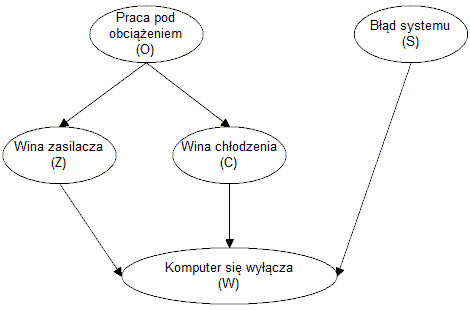
Przygotowanie sieci bayesowskiej umożliwiającej wspomaganie systemu diagnostycznego awarii komputera. Program obliczający współczynniki prawdopodobieństwa został napisany w języku Python 3. Wykorzystuje on implementację wnioskowania *Simple Bayesian Network*.

1. **Opis sieci bayesowskich**

Sieci bayesowskie reprezentują zależności między zdarzeniami. Konstruowane są za pomocą acyklicznych grafów skierowanych, w których zdarzenia są wierzchołkami i przyjmują wzajemnie wykluczające się wartości, a krawędzie reprezentują związki przyczynowe między nimi. Sieci te bazują na klasycznym rachunku prawdopodobieństwa , prawdopodobieństwie warunkowym oraz regułę Bayesa , których definicje są powszechnie dostępne.

1. **Schemat sieci bayesowskiej**

Poniżej znajduje się schemat prostej sieci diagnozującej przyczyny wyłączania się komputera w czasie jego pracy.



|  |  |
| --- | --- |
| **obciążenie (O)** | **P(O)** |
| tak | 0,3 |
| nie | 0,7 |

|  |  |  |
| --- | --- | --- |
| **obciążenie (O)** | **zasilacz (Z)** | **P(Z|O)** |
| tak | tak | 0,4 |
| tak | nie | 0,6 |
| nie | tak | 0,2 |
| nie | nie | 0,8 |

|  |  |  |
| --- | --- | --- |
| **obciążenie (O)** | **chłodzenie (C)** | **P(C|O)** |
| tak | tak | 0,8 |
| tak | nie | 0,2 |
| nie | tak | 0,1 |
| nie | nie | 0,9 |

|  |  |
| --- | --- |
| **system (S)** | **P(S)** |
| tak | 0,25 |
| nie | 0,75 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **zasilacz (Z)** | **chłodzenie (C)** | **system (S)** | **wyłączanie (W)** | **P(W|ZCS)** |
| tak | tak | tak | tak | 1 |
| tak | tak | tak | nie | 0 |
| tak | tak | nie | tak | 1 |
| tak | tak | nie | nie | 0 |
| tak | nie | tak | tak | 1 |
| tak | nie | tak | nie | 0 |
| nie | tak | tak | tak | 1 |
| nie | tak | tak | nie | 0 |
| nie | nie | tak | tak | 0,5 |
| nie | nie | tak | nie | 0,5 |
| nie | tak | nie | tak | 0,5 |
| nie | tak | nie | nie | 0,5 |
| tak | nie | nie | tak | 0,5 |
| tak | nie | nie | nie | 0,5 |
| nie | nie | nie | tak | 0,05 |
| nie | nie | nie | nie | 0,95 |

1. **Kod programu**

**from** bn **import** Factor

O = (**'O'**, [**'obciazenie'**, **'bezczynnosc'**])  
fO = Factor([O], [0.3, 0.7])  
  
Z = (**'Z'**, [**'zasilacz winny'**, **'zasilacz ok'**])  
fZ = Factor([O,Z], [0.4, 0.6,\  
 0.2, 0.8])  
  
C = (**'C'**, [**'chlodzenie winne'**, **'chlodzenie ok'**])  
fC = Factor([O,C], [0.8, 0.2,\  
 0.1, 0.9])  
  
S = (**'S'**, [**'system winny'**, **'system ok'**])  
fS = Factor([S], [0.25, 0.75])  
  
W = (**'W'**, [**'komputer sie wylacza'**, **'komputer ok'**])  
fW = Factor([Z,C,S,W], [1.0, 0.0,\  
 1.0, 0.0,\  
 1.0, 0.0,\  
 1.0, 0.0,\  
 0.5, 0.5,\  
 0.5, 0.5,\  
 0.5, 0.5,\  
 0.05, 0.95])

*# -------------------------------------------------------------------------***def** simple\_inference(query\_list, evidence\_list=[], \  
 factor\_multiply=fO\*fZ\*fC\*fS\*fW, \  
 variable\_list=[**'O'**, **'Z'**, **'C'**, **'S'**, **'W'**]):  
 **for** ev **in** evidence\_list:  
 factor\_multiply -= ev  
 **for** var **in** variable\_list:  
 **if** var **not in** query\_list:  
 factor\_multiply += var  
 factor\_multiply.normalize()  
 print(factor\_multiply)  
  
*# -------------------------------------------------------------------------*

1. **Działanie programu**

Uruchomić skrypt diagnoser.py oraz podać listę zapytań i dowodów.

Przykłady użycia:

print(**"Jakie jest prawdopodobieństwo awarii komputera?"**)  
query\_list, evidence\_list = [**'W'**], []  
simple\_inference(query\_list, evidence\_list)

W

f(komputer ok) 0.5523

f(komputer sie wylacza) 0.4477

print(**"Jakie jest prawdopodobieństwo awarii komputera w zależności od obciążenia?"**)  
query\_list, evidence\_list = [**'W'**, **'O'**], []  
simple\_inference(query\_list, evidence\_list)

O W

f(bezczynnosc, komputer ok) 0.4501

f(bezczynnosc, komputer sie wylacza) 0.2499

f(obciazenie, komputer ok) 0.1021

f(obciazenie, komputer sie wylacza) 0.1978

print(**"Jakie jest prawdopodobieństwo awarii komputera, jeśli zasilacz jest winny?"**)  
query\_list, evidence\_list = [**'W'**], [(**'Z'**, **'zasilacz winny'**)]  
simple\_inference(query\_list, evidence\_list)

W

f(komputer ok) 0.0

f(komputer sie wylacza) 1.0

print(**"Jakie jest prawdopodobieństwo awarii komputera, jeśli chłodzenie jest winne?"**)  
query\_list, evidence\_list = [**'W'**], [(**'Z'**, **'zasilacz winny'**)]  
simple\_inference(query\_list, evidence\_list)

W

f(komputer ok) 0.0

f(komputer sie wylacza) 1.0

print(**"Jakie jest prawdopodobieństwo awarii komputera, jeśli jest bezczynny i system jest winny?"**)  
query\_list, evidence\_list = [**'W'**], [(**'O'**, **'bezczynnosc'**), (**'S'**, **'system winny'**)]  
simple\_inference(query\_list, evidence\_list)

W

f(komputer ok) 0.4

f(komputer sie wylacza) 0.6

print(**"Jakie jest prawdopodobieństwo awarii komputera, jeśli jest obciążony i system jest winny?"**)  
query\_list, evidence\_list = [**'W'**], [(**'O'**, **'obciazenie'**), (**'S'**, **'system winny'**)]  
simple\_inference(query\_list, evidence\_list)

W

f(komputer ok) 0.3

f(komputer sie wylacza) 0.7

print(**"Jakie jest prawdopodobieństwo, że winny jest zasilacz?"**)  
query\_list, evidence\_list = [**'Z'**], []  
simple\_inference(query\_list, evidence\_list)

Z

f(zasilacz ok) 0.74

f(zasilacz winny) 0.26

print(**"Jakie jest prawdopodobieństwo, że komputer jest obciążony, jeśli winny jest zasilacz?"**)  
query\_list, evidence\_list = [**'O'**], [(**'Z'**, **'zasilacz winny'**)]  
simple\_inference(query\_list, evidence\_list)

O

f(bezczynnosc) 0.5385

f(obciazenie) 0.4615

print(**"Jakie jest prawdopodobieństwo, że winne jest chłodzenie?"**)  
query\_list, evidence\_list = [**'C'**], []  
simple\_inference(query\_list, evidence\_list)

C

f(chlodzenie ok) 0.69

f(chlodzenie winne) 0.31

print(**"Jakie jest prawdopodobieństwo, że winny jest system?"**)  
query\_list, evidence\_list = [**'S'**], []  
simple\_inference(query\_list, evidence\_list)

S

f(system ok) 0.75

f(system winny) 0.25

1. **Opis matematyczny**

Z opisu sieci bayesowskiej wynika, że znamy prawdopodobieństwo występowania obciążenia komputera P(O) = 0,3.

Znamy również rozkład prawdopodobieństw określających winę zasilacza w zależności od obciążenia P(Z|O). Aby obliczyć prawdopodobieństwo winy zasilacza, skorzystamy z twierdzenia o prawdopodobieństwie całkowitym:

Żeby obliczyć prawdopodobieństwo zdarzenia, że komputer jest obciążony, jeżeli wiemy, że to zasilacz jest winny, skorzystamy z reguły Bayesa:

Wartość ta pokrywa się z prawdopodobieństwem obliczonym przez program.

W przypadku ogólnym łączne prawdopodobieństwo dla całych rozkładów jest wyznaczane jako iloczyn poszczególnych rozkładów warunkowych.

1. **Podsumowanie**

Celem projektu było przygotowanie sieci bayesowskiej umożliwiającej wspomaganie systemu diagnostycznego awarii komputera. Napisany program umożliwia konstruowanie listy zapytań i dowodów oraz badanie rozkładów prawdopodobieństw dla tych danych.

Dodatkowo w trakcie tworzenia opisu, przybliżyliśmy prawa statystyki matematycznej, na których opiera się ten program oraz prześledziliśmy jego działanie na prostych zależnościach między zdarzeniami.