



Universidad
Rey Juan Carlos

Test modificaciones práctica 3

Inteligencia Artificial

Introducción

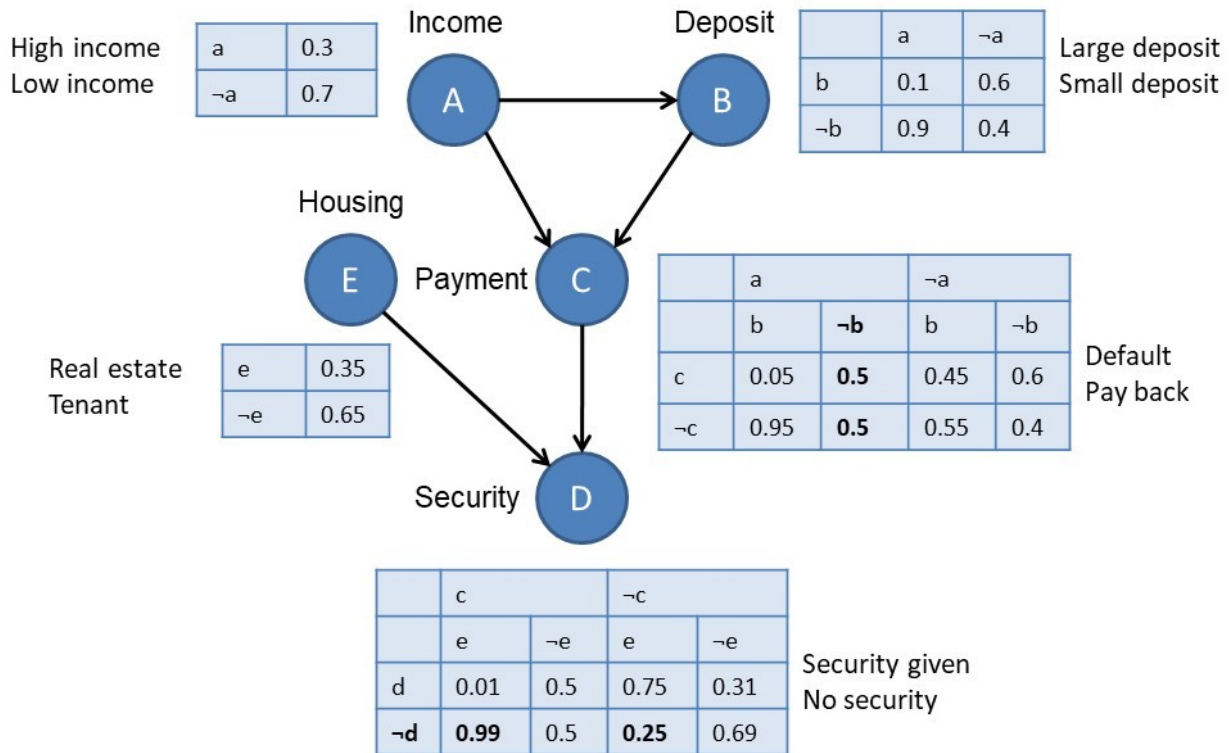
En este documento se proponen una serie de modificaciones al contenido de la práctica 3. Las modificaciones deben ir en el archivo `bayesAgents.py`, no es necesario hacer ningún cambio en otros ficheros.

Para completar la modificación, de debe añadir en la parte inferior del archivo `bayesAgents.py`, respetando el nombrado y después de la función `def combinations(n, r):`, las siguientes líneas:

```
def combinations(n, r):  
    ...  
  
    ...  
    Modifications  
    ...  
def constructInsuranceBayesNet():  
    """ MODIFICATION """  
    ...  
  
def inferenceOnInsuranceBayesNet(bayes_net):  
    """ MODIFICATION """  
    ...  
  
bayes_net = constructInsuranceBayesNet()  
print(inferenceOnInsuranceBayesNet(bayes_net))
```

1. Construir la estructura de la red Bayesiana del seguro

Define el método `constructInsuranceBayesNet()`. Construye la estructura de red Bayesiana descrita en la imagen inferior, incluyendo sus variables, valores, conexiones y CPTs. Esta función devolverá la red Bayesiana construída. Se incluye un fragmento de código que puede resultar útil tanto para esta pregunta como para la siguiente.



```
INCOME_VAR = "income"
HIGH_INCOME = "high_income"
LOW_INCOME = "low_income"
INCOME_VALS = [HIGH_INCOME, LOW_INCOME]

DEPOSIT_VAR = "deposit"
LARGE_DEPOSIT = "large_deposit"
SMALL_DEPOSIT = "small_deposit"
DEPOSIT_VALS = [LARGE_DEPOSIT, SMALL_DEPOSIT]

PAYMENT_VAR = "payment"
DEFAULT = "default"
PAY_BACK = "pay_back"
```

```
PAYMENT_VALS = [DEFAULT, PAY_BACK]

HOUSING_VAR = "housing"
REAL_STATE = "real_estate"
TENANT = "tenant"
HOUSING_VALS = [REAL_STATE, TENANT]

SECURITY_VAR = "security"
SECURITY_GIVEN = "security_given"
NO_SECURITY = "no_security"
SECURITY_VALS = [SECURITY_GIVEN, NO_SECURITY]
```

2. Hacer inferencia sobre la red Bayesiana del estudiante

Define el método `inferenceOnInsuranceBayesNet(bayes_net)`. Este método recibe la red Bayesiana del ejercicio anterior y realiza una inferencia por enumeración sobre la misma. Las evidencias son:

- La variable `INCOME_VAR` con un valor de `HIGH_INCOME`.
- La variable `HOUSING_VAR` con un valor de `TENANT`.

La variable que queremos consultar es `SECURITY_VAR`.

Este método devuelve el `Factor` resultante de realizar esta consulta.

Entregables

- Se crearán 2 carpetas (`sin_modificar`, `modificada`) en las que se incluirán los archivos de código Python `factorOperations.py`, `inference.py`, `bayesAgents.py`. En la primera irán los archivos que se han ido completando para la práctica a lo largo de las sesiones y en la segunda el archivo con las modificaciones propuestas en este documento.
- Ambas carpetas se comprimirán en un único archivo de nombre: `practica3_NombreApellidos.zip/tar.gz` que se entregará vía Aula Virtual.
- El código tiene que ir obligatoriamente comentado explicando su funcionalidad, incluyendo las modificaciones. Debe ser legible y estar debidamente tabulado.
- Se utilizarán sistemas anticopia y se podrá requerir explicación individual de la práctica en caso de duda.

- Entregar la práctica **sin modificar** y la práctica **modificada** con las propuestas de este documento.

Para comprimir las carpetas:

```
tar -czf practica3_NombreApellidos.tar.gz modificada/ sin_modificar/  
zip -r practica3_NombreApellidos.zip modificada/ sin_modificar/
```