CS486: Assignment 3 - Q1 a ~ e

factor.py

```
#!/usr/bin/python

class Factor(object):
	def __init__(self, variables, valueList, probabilities):
		self.valueTable = dict()
		for i in xrange(0,len(variables)):
			self.valueTable[variables[i]] = valueList[i]
		self.probabilities = probabilities

	def copy(self):
		newFactor = Factor([],[],[])
		newFactor.valueTable = dict(self.valueTable)
		newFactor.probabilities = list(self.probabilities)
		return newFactor

	# function that restricts a variable to some value in a given factor
	@staticmethod
	def restrict(factor, variable, value):
		newFactor = factor.copy()
		indiciesRestricted = list()
		restrictedFactorTable = list()
		restrictedProbabilities = list()
		index = 0
		for val in newFactor.valueTable[variable]:
			if val == value:
				indiciesRestricted.append(index)
			index += 1
		del newFactor.valueTable[variable]
		remainingVariables = newFactor.valueTable.keys()
		for var in remainingVariables:
			restrictedValueList = list()
			for i in xrange(0, len(newFactor.valueTable[var])):
				if i in indiciesRestricted:
					restrictedValueList.append(newFactor.valueTable[var][i])
			restrictedFactorTable.append(restrictedValueList)
		for i in xrange(0, len(remainingVariables)):
			newFactor.valueTable[remainingVariables[i]] = restrictedFactorTable[i]
		for i in xrange(0, len(newFactor.probabilities)):
			if i in indiciesRestricted:
				restrictedProbabilities.append(newFactor.probabilities[i])
		newFactor.probabilities = restrictedProbabilities
		return newFactor

	# function that multiplies two factors
	@staticmethod
	def multiply(factor1, factor2):
		factorVariables1 = factor1.valueTable.keys()
		factorVariables2 = factor2.valueTable.keys()
```

```
		commonVariables = list()
		newVariables = list()
		for var in factorVariables1:
			if var in factorVariables2:
				commonVariables.append(var)
				newVariables.append(var)
		for var in factorVariables1:
			if var not in newVariables:
				newVariables.append(var)
		for var in factorVariables2:
			if var not in newVariables:
				newVariables.append(var)
		newValueList = [[] for i in xrange(len(newVariables))]
		newProbabilities = list()
		for i in xrange(len(factor1.probabilities)):
			for j in xrange(len(factor2.probabilities)):
				isMatched = True
				for var in commonVariables:
					if factor1.valueTable[var][i] != factor2.valueTable[var][j]:
						isMatched = False
						break
				if isMatched:
					for k, var in enumerate(newVariables):
						if var in factorVariables1:
							newValueList[k].append(factor1.valueTable[var][i])
						else:
							newValueList[k].append(factor2.valueTable[var][j])
					newProbabilities.append(factor1.probabilities[i] * factor2.probabilities[j])
		return Factor(newVariables, newValueList, newProbabilities)

	# function that sums out a variable in a given factor
	@staticmethod
	def sumout(factor, variable):
		newFactor = factor.copy()
		if len(newFactor.valueTable.keys()) == 1:
			return newFactor
		variableDomain = list()
		for val in newFactor.valueTable[variable]:
			if val not in variableDomain:
				variableDomain.append(val)
		del newFactor.valueTable[variable]
		remainingVariables = newFactor.valueTable.keys()
		newValueList = [[] for i in xrange(len(remainingVariables))]
		newProbabilities = list()
		summedRows = list()
		for i in xrange(len(newFactor.probabilities)):
			if i in summedRows:
				continue
```

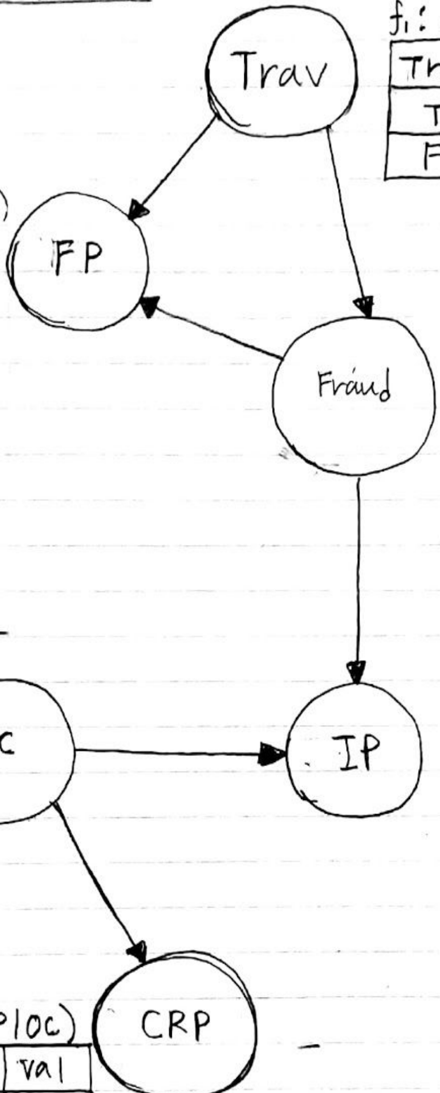
```
                        summedProbability = newFactor.probabilities[i]
                        for j in xrange(i+1, len(newFactor.probabilities)):
                                needSum = True
                                for var in remainingVariables:
                                        if newFactor.valueTable[var][i] !=
newFactor.valueTable[var][j]:
                                                needSum = False
                                if needSum and j not in summedRows:
                                        summedRows.append(j)
                                        summedProbability += newFactor.probabilities[j]
                        for k, var in enumerate(remainingVariables):
                                newValueList[k].append(newFactor.valueTable[var][i])
                        newProbabilities.append(summedProbability)
                return Factor(remainingVariables, newValueList, newProbabilities)

        # function that normalizes a factor by dividing each entry by the
        # sum of all the entries. This is useful when the factor is a distribution
        @staticmethod
        def normalize(factor):
                newFactor = factor.copy()
                totalSum = 0
                for prob in newFactor.probabilities:
                        totalSum += prob
                for i in xrange(len(newFactor.probabilities)):
                        newFactor.probabilities[i] = newFactor.probabilities[i]/totalSum
                return newFactor

        # function that computes P(queryVariables|evidenceList) by variable elimination
        @staticmethod
        def inference(factorList, queryVariables, orderedListOfHiddenVariables, evidenceList):
                newOrderedListOfHiddenVariables = filter(lambda x: x not in queryVariables and x
not in evidenceList.keys(), orderedListOfHiddenVariables)
                for evidence in evidenceList:
                        restrictedFactorList = list()
                        for factor in factorList:
                                if evidence in factor.valueTable.keys():
                                        # print
                                        print 'Restrict:',
                                        print evidence
                                        restrictedFactor = Factor.restrict(factor, evidence,
evidenceList[evidence])
                                        for key,val in restrictedFactor.valueTable.items():
                                                print key, "=>", val
                                        print restrictedFactor.probabilities
                                        restrictedFactorList.append(restrictedFactor)
                                else:
                                        restrictedFactorList.append(factor)
                        factorList = restrictedFactorList
                for hiddenVariable in newOrderedListOfHiddenVariables:
                        factorListToMultiply = list()
                        for factor in factorList:
```

```
			if hiddenVariable in factor.valueTable.keys():
				factorListToMultiply.append(factor)
		factorList = filter(lambda factor: factor not in factorListToMultiply, factorList)
		if len(factorListToMultiply) > 0:
			# print
			print 'Multiply:',
			print hiddenVariable
			productFactor = reduce(Factor.multiply, factorListToMultiply)
			for key,val in productFactor.valueTable.items():
				print key, "=>", val
			print productFactor.probabilities
		else:
			continue
		# print
		print 'Sumout:',
		print hiddenVariable
		factorSummedOut = Factor.sumout(productFactor, hiddenVariable)
		for key,val in factorSummedOut.valueTable.items():
			print key, "=>", val
		print factorSummedOut.probabilities
		factorList.append(factorSummedOut)
	# print
	print 'Multiply Query Variables:'
	if len(factorList) > 0:
		output = reduce(Factor.multiply, factorList)
		for key,val in output.valueTable.items():
			print key, "=>", val
		print output.probabilities
	# print
	print 'Normalize:'
	normalizedOutput = Factor.normalize(output)
	for key,val in normalizedOutput.valueTable.items():
		print key, "=>", val
	print normalizedOutput.probabilities
	return normalizedOutput
```

CS486: Assignment 3 - Q2 a

CS486 : Assignment 3

2.a) Bayes Network



f_1 : P(Trav)

Trav	val
T	0.05
F	0.95

f_2 : P(Fraud | Trav)

Fraud	Trav	val
T	T	0.01
T	F	0.004
F	T	0.99
F	F	0.996

f_3 : P(FP | Fraud, Trav)

FP	Fraud	Trav	val
T	T	T	0.9
T	T	F	0.1
T	F	T	0.9
T	F	F	0.01
F	T	T	0.1
F	T	F	0.9
F	F	T	0.1
F	F	F	0.99

f_4 : P(IP | Fraud, OC)

IP	Fraud	OC	val
T	T	T	0.02
T	T	F	0.011
T	F	T	0.01
T	F	F	0.001
F	T	T	0.98
F	T	F	0.989
F	F	T	0.99
F	F	F	0.999

f_5 : P(CRP | OC)

CRP	OC	val
T	T	0.1
T	F	0.001
F	T	0.9
F	F	0.999

f_6 : P(OC)

OC	val
T	0.6
F	0.4

<u>CS486: Assignment 3 - Q2 b~d</u>
*Results are rounded to the 4 decimal places
Q2(b)(I)
Query: P(Fraud)
Factors: f_1(Trav), f_2(Fraud,Trav)

Variable Elimination Step:

P(Fraud) $\propto$ f_7(Trav) = $\sum_{Trav} f_1(Trav)\, f_2(Fraud,Trav)$
Multiply f_1(Trav) and f_2(Fraud,Trav) with the variable Trav

Fraud	Trav	
T	T	0.0005
F	T	0.0495
T	F	0.0038
F	F	0.9462

Sumout the product with the variable Trav and normalize to f_7(Trav)

Fraud	
T	0.0043
F	0.9957

The prior probability that the current transaction is a fraud is:
P(Fraud = T) = 0.0043*100 = 0.43%

Q2(b)(II)
Query: P(Fraud|FP=T,IP=F,CRP=T)
Factors: f_1(Trav), f_2(Fraud,Trav), f_3(FP,Fraud,Trav), f_4(IP,Fraud,OC), f_5(CRP,OC), f_6(OC)

Variable Elimination Step:
-Restrict Step:

Restrict f_3(FP,Fraud,Trav) by the variable FP into f_7(Fraud,Trav)

Fraud	Trav	
T	T	0.9
T	F	0.1
F	T	0.9
F	F	0.01

Restrict f_4(IP,Fraud,OC) by the variable IP into f_8(Fraud,OC)

Fraud	OC	
T	T	0.98
T	F	0.989
F	T	0.99
F	F	0.999

Restrict f_5(CRP,OC) by the variable CRP into f_9(OC)

OC	
T	0.1
F	0.001

-Multiply&Sumout Step:

P(Fraud|FP=T,IP=F,CRP=T) $\propto$ f_{12}(Fraud) = f_{10}(Fraud) f_{11}(Fraud) = $\sum_{OC}$ f_6(OC) f_8(Fraud,OC) f_9(OC) $\sum_{Trav}$ f_1(Trav) f_2(Fraud,Trav) f_7(Fraud,Trav)

f_{10}(Fraud) = $\sum_{Trav}$ f_1(Trav) f_2(Fraud,Trav) f_7(Fraud,Trav)
Multiply f_1(Trav), f_2(Fraud,Trav), and f_7(Fraud,Trav) with the variable Trav

Fraud	Trav	
T	T	0.0005
F	T	0.0446
T	F	0.0004
F	F	0.0095

Sumout the product with the variable Trav to get f_{10}(Fraud)

Fraud	
T	0.0008
F	0.0540

f_{11}(Fraud) = $\sum_{OC}$ f_6(OC) f_8(Fraud,OC) f_9(OC)
Multiply f_6(OC), f_8(Fraud,OC), and f_9(OC) with the variable OC

Fraud	OC	
T	T	0.0588
T	F	0.0004
F	T	0.0594

F	F	0.0004

Sumout the product with the variable Trav to get f_{11}(Fraud)

Fraud	
T	0.0592
F	0.0598

f_{12}(Fraud) = f_{10}(Fraud) * f_{11}(Fraud)
Multiply Query Variables into f_{12}(Fraud)

Fraud	
T	4.9132348e-05
F	0.0032

-Normalize Step:

Fraud	
T	0.0150
F	0.9850

The probability that the current transaction is a fraud once we have verified that it is a foreign transaction, but not an internet purchase and that the card holder purchased computer related accessories in the past week is:

P(Fraud=T|FP=T,IP=F,CRP=T) = 0.0150 = 1.5%

Q2(c)

Query: P(Fraud|FP=T,IP=F,CRP=T,Trav=T)

Factors: f_1(Trav), f_2(Fraud,Trav), f_3(FP,Fraud,Trav), f_4(IP,Fraud,OC), f_5(CRP,OC), f_6(OC)

Variable Elimination Step:

-Restrict Step:

Restrict f_3(FP,Fraud,Trav) by the variable FP into f_7(Fraud,Trav)

Fraud	Trav	
T	T	0.9
T	F	0.1
F	T	0.9
F	F	0.01

Restrict f_4(IP,Fraud,OC) by the variable IP into f_8(Fraud,OC)

Fraud	OC	
T	T	0.98
T	F	0.989
F	T	0.99
F	F	0.999

Restrict f_5(CRP,OC) by the variable CRP into f_9(OC)

OC	
T	0.1
F	0.001

Restrict f_2(Fraud,Trav) by the variable Trav into f_{10}(Fraud)

OC	
T	0.01
F	0.99

Restrict f_7(Fraud,Trav) by the variable Trav into f_{11}(Fraud)

OC	
T	0.9
F	0.9

-Multiply&Sumout Step:

$P(Fraud|FP=T,IP=F,CRP=T,Trav=T) \propto f_{13}(Fraud) = f_{10}(Fraud)\ f_{11}(Fraud) \sum_{OC} f_8(Fraud,OC)\ f_9(OC)$

$f_{12}(Fraud,OC) = \sum_{OC} f_8(Fraud,OC)\ f_9(OC)$

Multiply $f_8(Fraud,OC)$ and $f_9(OC)$ with the variable OC

Fraud	OC	
T	T	0.0588
T	F	0.0004
F	T	0.0594
F	F	0.0004

Sumout the product with the variable OC to get $f_{12}(Fraud)$

Fraud	
T	0.0592
F	0.0598

$f_{13}(Fraud) = f_{10}(Fraud)\ f_{11}(Fraud)\ f_{12}(Fraud)$

Multiply Query Variables into $f_{12}(Fraud)$

Fraud	
T	2.6638020000000003e-05
F	0.0026640721800000005

-Normalize Step:

Fraud	
T	0.0099
F	0.9901

The probability that the current transaction is a fraud once we have verified that it is a foreign transaction, that the card holder purchased computer related accessories in the past week is, and the card holder is on a business trip, but not an internet purchase:

$P(Fraud=T|FP=T,IP=F,CRP=T,Trav=T) = 0.0099 = 0.99\%$

Q2(d)

-To simplify the problem, let's assume that I can only make a single action prior to the internet purchase to reduce the risk of fraud detection.
-Then, I can make a computer related purchase with internet prior to the week of the internet purchase via the stolen credit card in order to reduce the risk of fraud detection.
-To prove that this action could reduce the risk, let's first look at the query with CRP not known.

Query: P(Fraud|IP=T)
Factors: f_1(Trav), f_2(Fraud,Trav), f_3(FP,Fraud,Trav), f_4(IP,Fraud,OC), f_5(CRP,OC), f_6(OC)

Variable Elimination Step:
-Restrict Step:

Restrict f_4(IP,Fraud,OC) by the variable IP into f_7(Fraud,OC)

Fraud	OC	
T	T	0.02
T	F	0.011
F	T	0.01
F	F	0.001

-Multiply&Sumout Step:

P(Fraud|IP=T) $\propto$ f_{12}(Fraud) = f_9(Fraud) * f_{11}(CRP,Fraud) = $\sum_{CRP} \sum_{OC} f_5$(CRP,OC) f_6(OC) f_7(Fraud,OC) $\sum_{FP} \sum_{Trav} f_1$(Trav) f_2(Fraud,Trav) f_3(FP,Fraud,Trav)

f_8(FP,Fraud) = $\sum_{Trav} f_1$(Trav) f_2(Fraud,Trav) f_3(FP,Fraud,Trav)
Multiply f_1(Trav), f_2(Fraud,Trav) and f_3(FP,Fraud,Trav) with the variable Trav

FP	Fraud	Trav	
T	T	T	0.0005
F	T	T	5e-05
T	F	T	0.0446
F	F	T	0.0050
T	T	F	0.0004
F	T	F	0.0034
T	F	F	0.0095

F	F	F	0.9367

Sumout the product with the variable Trav to get f_8(FP,Fraud)

FP	Fraud	
T	T	0.0008
F	T	0.0035
T	F	0.0540
F	F	0.9417

$f_9(\text{Fraud}) = \sum_{FP} f_8(\text{FP},\text{Fraud})$

Multiply f_8(FP,Fraud) (actually no computation needed for the factor as there is only one factor to multiply) with the variable FP

FP	Fraud	
T	T	0.0008
F	T	0.0035
T	F	0.0540
F	F	0.9417

Sumout the product with the variable FP to get f_9(Fraud)

Fraud	
T	0.0043
F	0.9957

$f_{10}(\text{CRP},\text{Fraud}) = \sum_{OC} f_5(\text{CRP},\text{OC})\, f_6(\text{OC})\, f_7(\text{Fraud},\text{OC})$

Multiply f_5(CRP,OC), f_6(OC), and f_7(Fraud,OC) with the variable OC

CRP	Fraud	OC	
T	T	T	0.0012
F	T	T	0.0108
T	T	F	4.4e-06
F	T	F	0.0044
T	F	T	0.0006

F	F	T	0.0054
T	F	F	4e-07
F	F	F	0.0004

Sumout the product with the variable OC to get f_{10}(CRP,Fraud)

CRP	Fraud	
T	T	0.0012
F	T	0.0152
T	F	0.0006
F	F	0.0058

$f_{11}(Fraud) = \sum_{CRP} f_{10}(CRP,Fraud)$
Multiply f_{10}(Fraud) (actually no computation needed for the factor as there is only one factor to multiply) with the variable CRP

CRP	Fraud	
T	T	0.0012
F	T	0.0152
T	F	0.0006
F	F	0.0058

Sumout the product with the variable CRP to get f_{11}(Fraud)

Fraud	
T	0.0164
F	0.0064

$f_{12}(Fraud) = f_9(Fraud) * f_{11}(CRP,Fraud)$
Multiply Query Variables into f_{12}(Fraud)

Fraud	
T	7.052000000000001e-05
F	0.006372480000000001

-Normalize Step:

Fraud	

T	0.0109
F	0.9891

The probability that the current transaction is a fraud once we have verified that it is an internet purchase (the probability that the current transaction stops due to the fraud detection given it was an internet purchase):

$P(Fraud=T|IP=T) = 0.0109 * 100 = 1.09\%$

Query: $P(Fraud|IP=T,CRP=T)$

Factors: $f_1(Trav)$, $f_2(Fraud,Trav)$, $f_3(FP,Fraud,Trav)$, $f_4(IP,Fraud,OC)$, $f_5(CRP,OC)$, $f_6(OC)$

Variable Elimination Step:

-Restrict Step:

Restrict $f_4(IP,Fraud,OC)$ by the variable IP into $f_7(Fraud,OC)$

Fraud	OC	
T	T	0.02
T	F	0.011
F	T	0.01
F	F	0.001

Restrict $f_5(CRP,OC)$ by the variable CRP into $f_8(OC)$

OC	
T	0.1
F	0.001

-Multiply&Sumout Step:

$P(Fraud|IP=T) \propto f_{12}(Fraud) = f_{10}(Fraud) * f_{11}(Fraud) = \sum_{OC} f_6(OC)\ f_7(Fraud,OC)\ f_8(OC)$
$\sum_{FP} \sum_{Trav} f_1(Trav)\ f_2(Fraud,Trav)\ f_3(FP,Fraud,Trav)$

$f_9(FP,Fraud) = \sum_{Trav} f_1(Trav)\ f_2(Fraud,Trav)\ f_3(FP,Fraud,Trav)$

Multiply $f_1(Trav)$, $f_2(Fraud,Trav)$ and $f_3(FP,Fraud,Trav)$ with the variable Trav

FP	Fraud	Trav	
T	T	T	0.0005
F	T	T	5e-05
T	F	T	0.0446

F	F	T	0.0050
T	T	F	0.0004
F	T	F	0.0034
T	F	F	0.0095
F	F	F	0.9367

Sumout the product with the variable Trav to get f_9(FP,Fraud)

FP	Fraud	
T	T	0.0008
F	T	0.0035
T	F	0.0540
F	F	0.9417

$f_{10}(\text{Fraud}) = \sum_{FP} f_9(\text{FP,Fraud})$

Multiply f_9(FP,Fraud) (actually no computation needed for the factor as there is only one factor to multiply) with the variable FP

FP	Fraud	
T	T	0.0008
F	T	0.0035
T	F	0.0540
F	F	0.9417

Sumout the product with the variable FP to get f_{10}(Fraud)

Fraud	
T	0.0043
F	0.9957

$f_{11}(\text{CRP,Fraud}) = \sum_{OC} f_6(\text{OC})\ f_7(\text{Fraud,OC})\ f_8(\text{OC})$

Multiply f_6(OC), f_7(Fraud,OC), and f_8(OC) with the variable OC

Fraud	OC	
T	T	0.0012

T	F	4.4e-06
F	T	0.0006
F	F	4e-07

Sumout the product with the variable OC to get f_{11}(Fraud)

Fraud	
T	0.0012
F	0.0006

f_{12}(Fraud) = f_{10}(Fraud) * f_{11}(Fraud)
Multiply Query Variables into f_{12}(Fraud)

Fraud	
T	5.17892
F	0.0005978182799999999

-Normalize Step:

Fraud	
T	0.0086
F	0.9914

The probability that the current transaction is a fraud once we have verified that it is an internet purchase and we have made computer related purchase within a week (the probability that the current transaction stops due to the fraud detection given it was an internet purchase and we made the purchase on computer related product within a week):

P(Fraud=T|IP=T,CRP=T) = 0.0086 * 100 = 0.86%

Therefore, the probability of a fraud gets reduced by this specific action is:

1.09% - 0.86% = 0.23%