CS267A: Homework #6

Peter Racioppo (103953689)

May 25, 2020

Acknowledgements: If you have discussed with other students in the class regarding this homework, please acknowledge their names. See the syllabus for detailed policies about collaboration and academic honesty.

Problem 1

Assume we have a probabilistic database with four relations: R(r), S(s1,s2), U(u), and T(t). Compute each of the following queries symbolically using the lifted inference rules as we did in class, or state that it is not possible.

(1.) $\Pr(\exists x.R(x) \wedge T(x))$

Solution:
$\Pr(\exists x.R(x) \wedge T(x))$
= $1 - \Pr[\forall x.\neg R(x) \vee \neg T(x)]$
= $1 - \prod_i \Pr[\neg R(i) \vee \neg T(i)]$
= $1 - \prod_i (1 - \Pr[R(i) \wedge T(i)])$
= $1 - \prod_i (1 - \Pr[R(i)]\Pr[T(i)])$.

(2.) $\Pr(\exists x.\exists y.S(x,y) \wedge R(x))$

Solution:
$\Pr(\exists x.\exists y.S(x,y) \wedge R(x))$
= $1 - \Pr(\forall x.\forall y.\neg S(x,y) \vee \neg R(x))$
= $1 - \prod_i \Pr[\forall y.\neg S(i,y) \vee \neg R(i)]$
= $1 - \prod_i \Pr[\neg R(i) \vee \forall y.\neg S(i,y)]$
= $1 - \prod_i (1 - \Pr[R(i) \wedge \exists y.S(i,y)])$
= $1 - \prod_i (1 - \Pr[R(i)] \wedge \Pr[\exists y.S(i,y)])$
= $1 - \prod_i \{1 - \Pr[R(i)] \wedge (1 - \Pr[\forall y.\neg S(i,y)])\}$
= $1 - \prod_i \{1 - \Pr[R(i)] \wedge (1 - \prod_j \Pr[\neg S(i,j)])\}$
= $1 - \prod_i \{1 - \Pr[R(i)] \wedge (1 - \prod_j (1 - \Pr[S(i,j)]))\}$.

(3.) Pr$((\exists x.\exists y.R(x) \wedge S(x,y) \wedge T(y)) \vee (\exists x.U(x)))$

Solution:
Pr$([\exists x.\exists y.R(x) \wedge S(x,y) \wedge T(y)] \vee [\exists x.U(x)])$
= Pr$([\exists x.\exists y.R(x) \wedge S(x,y) \wedge T(y)])$+Pr$([\exists x.U(x)])-Pr([\exists x_1.\exists y.R(x_1) \wedge S(x_1,y) \wedge T(y)] \wedge [\exists x_2.U(x_2)])$.
Lifted inference fails on the first clause.
NOT POSSIBLE.

(4.) Pr$(\exists x_1.\exists x_2.\exists y_1.\exists y_2.R(x_1) \wedge S(x_1,y_1) \wedge T(x_2) \wedge S(x_2,y_2))$

Solution:
Pr$(\exists x_1.\exists x_2.\exists y_1.\exists y_2.R(x_1) \wedge S(x_1,y_1) \wedge T(x_2) \wedge S(x_2,y_2))$
= 1$-$ Pr$(\forall x_1.\forall x_2.\forall y_1.\forall y_2.\neg R(x_1) \vee \neg S(x_1,y_1) \vee \neg T(x_2) \vee \neg S(x_2,y_2))$
= 1$-$ Pr$([\forall x_1.\forall y_1.\neg R(x_1) \vee \neg S(x_1,y_1)] \vee [\forall x_2.\forall y_2.\neg T(x_2) \vee \neg S(x_2,y_2)])$
= 1$-$ Pr$([\forall x_1.\forall y_1.\neg R(x_1) \vee \neg S(x_1,y_1)] \vee [\forall x_1.\forall y_1.\neg T(x_1) \vee \neg S(x_1,y_1)])$
= 1$-$ Pr$(\forall x_1.\forall y_1.[\neg R(x_1) \vee \neg S(x_1,y_1)] \vee [\neg T(x_1) \vee \neg S(x_1,y_1)])$
= 1$-$ Pr$(\forall x.\forall y.\neg R(x) \vee \neg S(x,y) \vee \neg T(x))$
= $1 - \prod_i$Pr$(\forall y.\neg R(i) \vee \neg S(i,y) \vee \neg T(i))$
= $1 - \prod_i$Pr$(\neg R(i) \vee \neg T(i) \vee \forall y.\neg S(i,y))$
= $1 - \prod_i[1-$Pr$(R(i) \wedge T(i) \wedge \exists y.S(i,y))]$
= $1 - \prod_i[1-$Pr$(R(i))\times$Pr$(T(i))\times$Pr$(\exists y.S(i,y))]$
= $1 - \prod_i[1-$Pr$(R(i))\times$Pr$(T(i)) \times [1-$Pr$(\forall y.\neg S(i,y))]]$
= $1 - \prod_i[1-$Pr$(R(i))\times$Pr$(T(i)) \times (1 - \prod_jPr(\neg S(i,j)))]$
= $1 - \prod_i[1-$Pr$(R(i))\times$Pr$(T(i)) \times (1 - \prod_j[1-Pr(S(i,j))])]$

(5.) Pr$((\exists x_1.\exists y_1.R(x_1) \wedge S(x_1,y_1)) \vee (\exists x_2.\exists y_2.T(y_2) \wedge S(x_2,y_2))$

Solution:
Pr$((\exists x_1.\exists y_1.R(x_1) \wedge S(x_1,y_1)) \vee (\exists x_2.\exists y_2.T(y_2) \wedge S(x_2,y_2))$
= Pr$((\exists x_1.R(x_1) \wedge \exists y_1.S(x_1,y_1)) \vee (\exists y_2.T(y_2) \wedge \exists x_2.S(x_2,y_2))$
= Pr$((\exists x.R(x) \wedge \exists y.S(x,y)) \vee (\exists y.T(y) \wedge \exists x.S(x,y))$
= 1$-$Pr$((\forall x.\neg R(x) \vee \forall y.\neg S(x,y)) \wedge (\forall y.\neg T(y) \vee \forall x.\neg S(x,y))$
= 1$-$Pr$([(\forall x.\neg R(x) \vee \forall y.\neg S(x,y)) \wedge \forall y.\neg T(y)] \vee [(\forall x.\neg R(x) \vee \forall y.\neg S(x,y)) \wedge \forall x.\forall y.\neg S(x,y)])$
= 1$-$Pr$([\forall x.\forall y.\neg R(x) \wedge \neg T(y) \vee \neg S(x,y) \wedge \neg T(y)] \vee [\forall x.\forall y.\neg R(x) \wedge \neg S(x,y) \vee \neg S(x,y)])$
= 1$-$Pr$(\forall x.\forall y.[\neg R(x) \wedge \neg T(y) \vee \neg S(x,y) \wedge \neg T(y)] \vee [\forall x.\forall y.\neg S(x,y)])$
= 1$-$Pr$(\forall x.\forall y.\neg R(x) \wedge \neg T(y) \vee \neg S(x,y) \wedge \neg T(y) \vee \neg S(x,y))$
= 1$-$Pr$(\forall x.\forall y.\neg R(x) \wedge \neg T(y) \vee \neg S(x,y))$
= 1$-$Pr$(\forall x.\forall y.\neg R(x) \vee \neg S(x,y) \wedge \neg T(y) \vee \neg S(x,y))$
= 1$-$Pr$(\forall x.\forall y.\neg R(x) \vee \neg S(x,y))$+Pr$(\forall x.\forall y.\neg T(y) \vee \neg S(x,y))-Pr((\forall x.\forall y.\neg R(x) \vee \neg S(x,y)) \vee (\neg T(y) \vee \neg S(x,y)))$
= 1$-$Pr$(\forall x.\forall y.\neg R(x) \vee \neg S(x,y))$+Pr$(\forall x.\forall y.\neg T(y) \vee \neg S(x,y))-Pr((\forall x.\forall y.\neg R(x) \vee \neg T(y) \vee \neg S(x,y)))$
Lifted inference fails on the last clause, and there are no possible cancellations.
NOT POSSIBLE.

(6.) $\Pr((\exists x_1.\exists y_1.R(x_1) \wedge S(x_1, y_1)) \vee (\exists x_2.\exists y_2.S(x_2, y_2) \wedge T(y_2)) \vee (\exists x_3.\exists y_3.R(x_3) \wedge T(y_3))$
(Hint: you may need to symbolically cancel some queries)

Solution:
$\Pr((\exists x_1.\exists y_1.R(x_1) \wedge S(x_1, y_1)) \vee (\exists x_2.\exists y_2.S(x_2, y_2) \wedge T(y_2)) \vee (\exists x_3.\exists y_3.R(x_3) \wedge T(y_3)))$
$= \Pr((\exists x.\exists y.R(x) \wedge S(x, y)) \vee (\exists x.\exists y.S(x, y) \wedge T(y)) \vee (\exists x.\exists y.R(x) \wedge T(y)))$
$= \Pr(R(x) \wedge S(x, y)) + \Pr(S(x, y) \wedge T(y)) + \Pr(R(x) \wedge T(y)) - \Pr((R(x) \wedge S(x, y)) \wedge (S(x, y) \wedge T(y))) - \Pr((R(x) \wedge S(x, y)) \wedge (R(x) \wedge T(y))) - \Pr((S(x, y) \wedge T(y)) \wedge (R(x) \wedge T(y))) + \Pr((R(x) \wedge S(x, y)) \wedge (S(x, y) \wedge T(y)) \wedge (R(x) \wedge T(y)))$
$= \Pr(R(x) \wedge S(x, y)) + \Pr(S(x, y) \wedge T(y)) + \Pr(R(x) \wedge T(y)) - \Pr(R(x) \wedge S(x, y) \wedge T(y)) - \Pr(R(x) \wedge S(x, y) \wedge T(y)) - \Pr(S(x, y) \wedge T(y) \wedge R(x)) + \Pr(R(x) \wedge S(x, y) \wedge T(y))$
All of the four last terms are the same, except for the last term, which has the opposite sign. Thus, two terms cancel and we are left with:
$= \Pr(R(x) \wedge S(x, y)) + \Pr(S(x, y) \wedge T(y)) + \Pr(R(x) \wedge T(y)) - 2 \times \Pr(R(x) \wedge S(x, y) \wedge T(y))$
Lifted inference fails on the last term, so the cancellation has not helped us.
NOT POSSIBLE.

Problem 2

Recall the following query from class: $H_0 = \exists x.\exists y.S(x) \wedge F(x, y) \wedge R(y)$.
In class, we showed that evaluating H_0 for an arbitrary database is #P-hard in the size of the database. We proved this by reduction to the positive partitioned 2-CNF (#PP2CNF) counting problem, for which we give a formal definition.
Definition 1. *A 2-CNF is a CNF where each clause has exactly 2 literals. A positive partitioned 2-CNF is a 2-CNF where variables are partitioned into 2 sets X, Y, $X \cap Y = \emptyset$, and every clause is of the form $(x \vee y)$ with $x \in X$ and $y \in Y$. Finally, #PP2CNF is the problem of counting how many models a PP2CNF formula has.*
For each of the following PP2CNF formulae, give three tables S(x), F(x, y), and R(y) such that evaluating H_0 on these tables can be used to compute the model count of the formula:

1. $f = x_1 \vee y_1$. Hint: What is the model count for this formula? Since you can compute it by hand, use it to test your answer.

Solution:
The model count for this formula is 3.
Consider $\neg H_0 = \forall x.\forall y.\neg S(x) \vee \neg F(x, y) \vee \neg R(y)$
Let $F(x, y) = 1, \forall x, \forall y$. Then, $\neg H_0 = \forall x.\forall y.\neg S(x) \vee \neg R(y)$.
Each entry of $\neg S(x)$ and $\neg R(y)$ should have probability 0.5, so each entry of $S(x)$ and $R(y)$ should have probability $1 - 0.5 = 0.5$. Then, the model count is $K = P(\neg H_0)/p = (1 - P(H_0))/p$.
In this case, $K = (1 - 1/4)/(1/4) = (3/4)/(1/4) = 3$.

2. $f = (x_1 \vee y_1) \wedge (x_2 \vee y_2) \wedge (x_3 \vee y_3) \wedge (x_1 \vee y_3)$.
Hint: Represent the x_i variables as tuples in table S(x), the y_i variables as tuples in table R(y), and each clause as a tuple in F(x,y). How do you relate “a variable is true” to the presence of a tuple in S or R? How do you relate “a clause is true” to the presence of a tuple in F? How can you choose the weights of each tuple so that the query can be used to compute (but does not necessarily equal) the model count?

Solution:
Consider $\neg H_0 = \forall x. \forall y. \neg S(x) \vee \neg F(x,y) \vee \neg R(y)$ and $\neg f = (\neg x_1 \wedge \neg y_1) \vee (\neg x_2 \wedge \neg y_2) \vee (\neg x_3 \wedge \neg y_3) \vee (\neg x_1 \wedge \neg y_3)$.

Let $\neg F(x,y) = 0$ if $(x = \neg x_1, y = \neg y_1), (x = \neg x_2, y = \neg y_2), (x = \neg x_3, y = \neg y_3)$, or $(x = \neg x_1, y = \neg y_3)$ and $\neg F(x,y) = 1$ otherwise. Thus, $\neg H_0 = \forall x. \forall y. \neg S(x) \vee \neg R(y)$ if and only if both variables occur in one of the four clauses.

There are 6 binary variables, so there are $2^6 = 64$ possible worlds. The variables $\neg x_1$ and $\neg y_3$ occur 3/2 as often as the variables $\neg x_2, \neg x_3, \neg y_1, \neg y_2$, so their weights should be 3/2 as large.

Let p_i be the weight of $\neg x_i$ and q_i be the weight of $\neg y_i$, $i \in [1, 2, 3]$.
Set $P(\neg H_0) = p_1 q_1 + p_2 q_2 + p_3 q_3 + p_1 q_3 = P(\neg f)$.
Then, setting $p_1 = q_3 = \frac{3}{2} p_2 = \frac{3}{2} p_3 = \frac{3}{2} q_1 = \frac{3}{2} q_2$ fully defines all six variables, and the model count is given by $K = 64 - \neg K = 64 - P(\neg H_0)/p$ with $p = \frac{1}{64}$.

Problem 3

It is well-known that solving the satisfiability problem is NP-complete. Suppose we show that a SAT-solver can be used to solve a Sudoku puzzle. Does this mean that solving Sudoku puzzles is NP-hard? Why or why not?

Solution:
NO.
Showing that a SAT-solver can be used to solve a Sudoku puzzle would ***not*** show that solving Sudoku puzzles is NP-hard. In order to show that solving Sudoku puzzles is NP-hard, we'd need to show that a problem in NP reduces to solving Sudoku, since in this case Sudoku would be at least as hard as a problem in NP. In this case, we've done the opposite, showing that Sudoku can be reduced to an NP-complete problem.

Problem 4

The #PP3CNF counting problem is defined as follows:
Definition 2. *A 3-CNF is a CNF where each clause has exactly 3 literals. A positive partitioned 3-CNF is a 3-CNF whose variables, labeled V, are partitioned into 3 disjoint sets X, Y, Z, i.e.: $V = X \cup Y \cup Z, X \cap Y = Y \cap Z = X \cap Z = \emptyset$ and each clause contains exactly one positive literal from X, Y, and Z. Finally, #PP3CNF is the problem of counting the number of satisfying assignments (models) of a PP3CNF formula. Is the #PP3CNF problem #P-hard? Explain why or why not.*

Problem 4 continued on next page. . .

Solution:
YES. The #PP3CNF counting problem can be reduced to the #PP2CNF.
In particular, consider a PP2CNF whose variables V_1 are partitioned into 2 disjoint sets X and Y, i.e. $V_1 = X \cup Y, X \cap Y = \emptyset$. Let $Z = \emptyset$ and consider the PP3CNF whose variables V_2 are partitioned into 3 disjoint sets X, Y, and Z, i.e. $V_2 = X \cup Y \cup Z, X \cap Y = Y \cap Z = X \cap Z = \emptyset$. Note that the condition that $Y \cap Z = X \cap Z = \emptyset$ is satisfied for any X, Y. Furthermore, $V_2 = X \cup Y = V_1$. Thus, solving this PP3CNF is equivalent to solving the PP2CNF. Since the #PP2CNF problem is #P-hard, it follows that the #PP3CNF is #P-hard as well.