Cancer/Smoke/Friends: a classical example for MLNs

This simple DeepDive example is based on a classical *Markov Logic Networks* example to show probabilistic inference and factor graphs functionalities of DeepDive. You can read more about probabilistic inference and factor graphs in our detailed documentation.

Inference rules

The objectives in this example are to infer whether a person smokes, and whether a person has cancer with a some probability using the factor `A => B`. Here, `A => B` reads "`A` implies `B`", meaning that if `A` is true, then `B` is also true.

We introduce two rules:

1. If person A smokes, then A might have a cancer.
2. If two people A and B are friends and A smokes, then B might also smoke.

These rules can be written in DDlog as follows (in the `app.ddlog` file):

```
person (
    person_id bigint,
    name text
    ).

person_has_cancer? (
    person_id bigint
).

person_smokes? (
    person_id bigint
).

friends (
    person_id bigint,
    friend_id bigint
).

@weight(0.5)
person_smokes(p) => person_has_cancer(p) :-
    person(p, _).

@weight(0.4)
person_smokes(p1) => person_smokes(p2) :-
    person(p1, _), person(p2, _), friends(p1, p2).
```

Setup

Before running the example, please check that DeepDive has been properly installed and the necessary files (`app.ddlog`, `db.url`, and deepdive.conf) and directories (input/) associated with this example are stored in the current working directory. Input directory should have the data files (`friends.tsv`, `person_has_cancer.tsv`, `person_smokes.tsv`, and `person.tsv`). In order to use DeepDive, a database instance must be running to accept requests, and the database location must be specified in the `db.url`. You can refer to the tutorial for further detail.

Running

Now you are ready to run the example. First, you have to compile the code using the following command.

```
deepdive compile
```

Once it has compiled with no error, you can run the following command to see the list of deepdive targets.

```
deepdive do
```

To run the entire pipeline you can run the following command.

```
deepdive run
```

This will display a plan for deepdive to run your pipeline. To start the pipeline, exit the editor with `:wq` command.

Results

Once the pipeline has completed running, you can view the results in the database using SQL or DDlog queries. The entire database should look like this:

```
                                        List of relations

 Schema |                           Name                            | Ty
pe   | Owner |     Size     | Description

--------+-----------------------------------------------------------
-+-------+-------+------------+-------------

 public | dd_graph_variables_holdout                                |
table | user | 0 bytes    |

 public | dd_graph_variables_observation                            |
table | user | 0 bytes    |

 public | dd_graph_weights                                          | v
iew   | user | 0 bytes    |

 public | dd_inference_result_variables                             |
table | user | 8192 bytes |

 public | dd_factors_inf_imply_person_smokes_person_has_cancer
| table | user | 8192 bytes |

 public | dd_factors_inf_imply_person_smokes_person_smokes
| table | user | 8192 bytes |

 public | dd_weights_inf_imply_person_smokes_person_has_cancer
| table | user | 16 kB      |

 public | dd_weights_inf_imply_person_smokes_person_smokes
| table | user | 16 kB      |

 public | friends                                                   | ta
ble | user | 8192 bytes |

 public | person                                                    | ta
ble | user | 16 kB      |

 public | person_has_cancer                                         | t
able | user | 8192 bytes |

 public | person_has_cancer_calibration                             |
view  | user | 0 bytes    |

 public | person_has_cancer_inference                               |
view  | user | 0 bytes    |
 public | person_smokes                                           | t
able | user | 8192 bytes |

 public | person_smokes_calibration                               |
view  | user | 0 bytes    |

 public | person_smokes_inference                                 | v
iew  | user | 0 bytes    |

(16 rows)
```

Tables `person`, `friends`, `person_has_cancer`, and `person_smokes` hold the input data we prepared under the `input/` directory. To see what DeepDive inferred from our data, you can look at `person_smokes_inference` and `person_has_cancer_inference`. The two views should look like the following:

```
deepdive sql "SELECT * FROM person_smokes_inference"
 person_id | dd_id | label | category | expectation
-----------+-------+-------+----------+-------------
         4 |     9 |       |        1 |       0.643
         2 |     7 |       |        1 |       0.506
         6 |    11 |       |        1 |       0.468
         5 |    10 |       |        1 |       0.451
(4 rows)


deepdive sql "SELECT * FROM person_has_cancer_inference"
 person_id | dd_id | label | category | expectation
-----------+-------+-------+----------+-------------
         3 |     2 |       |        1 |       0.635
         1 |     0 |       |        1 |       0.614
         6 |     5 |       |        1 |        0.57
         2 |     1 |       |        1 |       0.563
         4 |     3 |       |        1 |       0.563
         5 |     4 |       |        1 |       0.551
(6 rows)
```

The `dd_id` column is for internal usage and can be ignored by the user and `person_id` is the user defined identifier in the input data. You can see that DeepDive uses the given data and inference rules to predict the probability of the person being a smoker or having cancer in the `expectation` column.