Report for Lab 1 [CS5340]

Wu, Yihang A0285643W

Apart from functions listed in lab1 instruction pdf, some auxiliary functions were added for support some functions.

compute_marginals_bp

For function `compute_marginals_bp(V, factors, evidence)`, I added:

- `collect(g,i,j,msg)`
- `sendmessage(g,j,i,msg)`
- `distribute(g,i,j,msg)`
- `computeMarginal(g,i,msg)`

The design of these functions is according to Lecture 4 slides Page. 68.

map_eliminate

For function `map_eliminate`, I added:

- `map_sendmsg(g,j,i,msg,cfg)`: In this function, the operation for passing messages is the same as operations in `collect(g,i,j,msg)`, and original `factor_product` and `factor_marginalize` is replaced by `factor_sum` and `factor_max_marginalize` to get $m_{ji}^{max}(x_i)$. As `factor_max_marginalize` return factor with argmax configuration on unmarginalized vars, we can then get the configuration $\delta_{ji}(x_i)$ (marginalized vars) by replacing the keys in argmax dictionary:

  ```
  cfg[j][i] = {}
      for idx, dict_item in enumerate(msg[j][i].val_argmax):
          cfg[j][i][idx] = list(dict_item.values())[0]
  ```

  It's very convenient.

- `map_collect(g,i,j,msg,cfg)`
- `map_distribute(g, i, j, msg, cfg, max_decoding)`

These functions' pipelines have some similarities with the 4 functions I implemented above for `compute_marginals_bp`, and some modifications are made to make them work in log spaces and to find the configuration of max probability. The design of these functions is according to Lecture 5 slides Page. 48. But, there are some little changes. There is no `SETVALUE(i,j)` function like pseudo codes in slides. Considering it can be realized in one line code, and only be called in `map_distribute` function, I realized it in `map_distribute` function.