Introduction

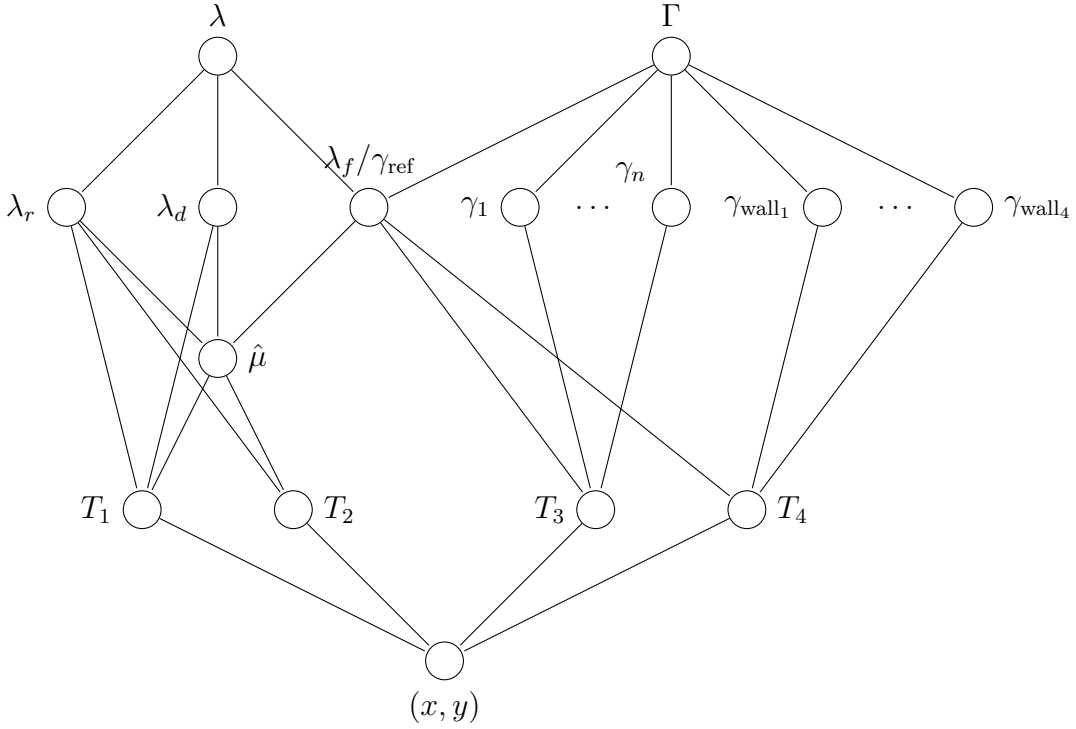
Let Γ represent the knowledge of the table. Γ is composed of $\gamma_1, \ldots, \gamma_n$ corresponding to the objects on the table and $\gamma_{\text{wall}_1}, \gamma_{\text{wall}_2}, \gamma_{\text{wall}_3}, \gamma_{\text{wall}_4}$ corresponding to the four walls (edges) of the table.

Let λ be the command the user gives, consisting of a distance λ_d a direction (relation) λ_r and a reference object λ_f. For the sake of testing this algorithm, we will assume perfect ability to parse the command. Therefore we assume there is a perfect correspondence between λ_f and γ_{ref}, where $\gamma_{\text{ref}} \in \{\gamma_1, \ldots, \gamma_n\}$. In addition, for the model not including rotation, we assume that the direction is in the set {left, right, in front, behind} and map λ_r to the corresponding direction vectors $\{[1,0],[-1,0],[0,1],[0,-1]\}$

As an example, if the command is 'five inches to the right of the bowl', then we have $\lambda_d = 5$, $\lambda_r = [1,0]$, $\lambda_f =$ the bowl.

Our goal then is to estimate a function $\mathbb{P}(x, y|\lambda, \Gamma)$ which gives the probability that a user was referring to the point (x, y) given the command λ and world Γ.

From the command and reference, we calculate a naive mean, which is the point you would select if you went exactly the distance specified by the command. From this we calculate four features for a log-linear model. The four features are explained below. All the terms so far described are related to each other via the following graphical model:



Feature Calculation

0.1 Calculating $\hat{\mu}$

The naive mean, $\hat{\mu}$, is what is obtained by going exactly the distance specified in the command from the edge of the object. This is calculated as follows:

$$\hat{\mu} = \begin{cases} \gamma_{\text{ref}}.center + \frac{\gamma_{\text{ref}}.height}{2} + \lambda_d, & \lambda_r = [0, -1] \\ \gamma_{\text{ref}}.center - \frac{\gamma_{\text{ref}}.height}{2} - \lambda_d, & \lambda_r = [0, 1] \\ \gamma_{\text{ref}}.center + \frac{\gamma_{\text{ref}}.width}{2} + \lambda_d, & \lambda_r = [1, 0] \\ \gamma_{\text{ref}}.center - \frac{\gamma_{\text{ref}}.width}{2} - \lambda_d, & \lambda_r = [-1, 0] \end{cases}$$

0.2 Calculating T_1 and T_2

We use three assumptions about the data here. First, that the data are distributed in a gaussian manner. Second, that the variance in the direction of the command (i.e. in the x direction for 'left' or 'right' and the y direction for 'in front' and 'behind') is independent of the variance in the orthogonal direction. Third, that variance in the direction of the command scales linearly with the distance of the command, while variance in the orthogonal direction is constant.

From this, our goal is to generate features that tell us about the probability of a point (x, y) given λ, Γ. Let $v = (x, y) - \hat{\mu}$. A gaussian version of this probability incorporating the above assumptions would be:

$$\frac{1}{Z} \exp\left(\frac{\langle v, \lambda_r \rangle^2}{k_1 \lambda_d}\right) \exp\left(\frac{[v - \langle v, \lambda_r \rangle \lambda_r]^2}{k_2}\right)$$

Turning these into features in a log-linear distribution, we then get:

$$T_1(x, y|\lambda, \Gamma) = \frac{1}{\lambda_d} \langle v, \lambda_r \rangle^2$$
$$T_2(x, y|\lambda, \Gamma) = [v - \langle v, \lambda_r \rangle \lambda_r]^2$$

with $v = (x, y) - \hat{\mu}$ as before.

0.3 Calculating T_3

****** TBD ********

0.4 Calculating T_4

0.5 Putting it Together

0.6 MLE Estimation for Learning Weights

OLD STUFF

0.7 Object Distance Estimation

This is a distribution that penalizes estimated locations that are closer to a different object in the world than they are to the reference object. It takes the form of an exponential distribution. The parameter was found via grid search.

$$p(x, y|command, world) = \frac{1}{2.7} e^{\frac{1}{2.7}(||(x,y)-(x_{ref},y_{ref})||-\min_{obj \in world} ||(x,y)-(x_{obj},y_{obj})||)}$$

0.8 Wall Distance Estimation

This is a distribution that penalizes estimated locations that are closer to a wall than they are to the reference object. It takes the form of an exponential distribution. The parameter was found via grid search. Note: edges of tables count as walls.

$$p(x, y|command, world) = \frac{1}{1.2} e^{\frac{1}{1.2}(||(x,y)-(x_{ref},y_{ref})||-\min_{walls}||(x,y)-(x_{wall},y_{wall})||}$$