Supplementary Material For
PlanIt: A Crowdsourcing Approach for Learning to Plan Paths from Large Scale Preference Feedback

Ashesh Jain, Debarghya Das, Jayesh K. Gupta and Ashutosh Saxena

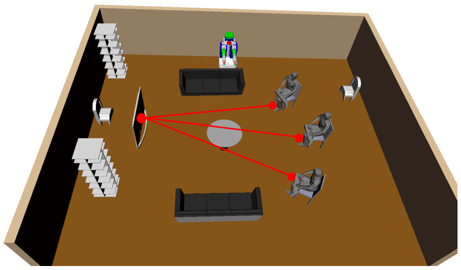

Fig. 1: An environment with three instances of watching activity.

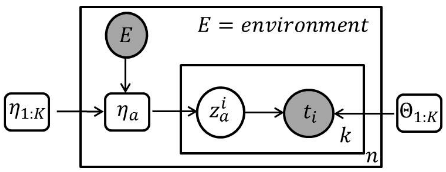

Fig. 2: Generative process for modeling the user preference data.

I. Generative Model: Learning the Parameters

Given user preference data from PlanIt, we learn the model parameters. Since our goal was to make the data collection easier for users, the labels we get are either bad, neutral or good for a particular segment of the video. The challenge is that we do not know which activity a is being affected by a given waypoint t_i during feedback. A waypoint could even be influencing multiple activities. For example, in Fig. 1 a waypoint passing between the human and TV could affect multiple watching activities.

We therefore define a latent random variable $z_a^i \in \{0, 1\}$ for waypoint t_i, such that $p(z_a^i|E)$ (or η_a) is the (prior) probability of user data arising from activity a. Incorporating this parameter gives the following cost function:

$$\mathbf{\Psi}(\{t_1, .., t_k\}|E) = \prod_{i=1}^{k} \underbrace{\sum_{a \in \mathcal{A}_E} p(z_a^i|E)\ \Psi_a(t_i|E)}_{\text{Marginalizing latent variable } z_a^i} \quad (1)$$

where $\mathcal{A}_E$ is the set of activities in environment E.[1] Figure 2 shows the generative process for preference data.

Training data: We obtain user preferences over n environments $E_1, .., E_n$. For each environment E we consider m trajectory segments $\mathcal{T}_{E,1}, .., \mathcal{T}_{E,m}$ labeled as bad by users. For each segment $\mathcal{T}$ we sample k waypoints $\{t_{\mathcal{T},1}, .., t_{\mathcal{T},k}\}$. We use $\Theta \in \mathbb{R}^{30}$ to denote the model parameters and solve the following maximum likelihood problem:

$$\begin{aligned}\Theta^* &= \arg\max_{\Theta} \prod_{i=1}^{n}\prod_{j=1}^{m} \mathbf{\Psi}(\mathcal{T}_{E_i,j}|E_i;\Theta)\\ &= \arg\max_{\Theta} \prod_{i=1}^{n}\prod_{j=1}^{m}\prod_{l=1}^{k}\sum_{a\in\mathcal{A}_{E_i}} p(z_a^l|E_i;\Theta)\ \Psi_a(t_{\mathcal{T}_{E_i,j},l}|E_i;\Theta)\end{aligned} \quad (2)$$

Eq. (2) does not have a closed form solution. We follow Expectation-Maximization (EM) procedure to learn the model parameters. In E-step, we calculate the posterior activity assignment $p(z_a^l|t_{\mathcal{T}_{E_i,j},l}, E_i)$ for all the waypoints and update the parameters in the M-step.

E-step: In this step keeping the model parameters fixed we find the posterior probability of a waypoint t affecting an activity a.

$$p(z_a|t, E;\Theta) = \frac{p(z_a|E;\Theta)\Psi_a(t|E;\Theta)}{\sum_{a\in\mathcal{A}_E} p(z_a|E;\Theta)\Psi_a(t|E;\Theta)} \quad (3)$$

We calculate this posterior for every waypoint t in our data.

M-step: Using the posterior from E-step we update the model parameters in this step. Our affordance representation consists of three distributions, namely: Gaussian, von-Mises and Beta. The parameters of Gaussian, and mean (μ) of von-Mises are updated in a closed form. Following Sra [2] we perform first order approximation to update the variance (κ) of von-Mises. The parameters of beta distribution (α and β) are approximated using first and second order moments of the data.

Estimating von-Mises distribution parameters: von-Mises is parameterized by a scalar mean μ and variance κ. Mean for an activitiy a has closed form update expression:

$$\mu_a = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{l=1}^{k} p(z_a^l|t_{\mathcal{T}_{E_i,j},l}, E_i)\mathbf{x}_{t_{\mathcal{T}_{E_i,j},l}}}{\|\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{l=1}^{k} p(z_a^l|t_{\mathcal{T}_{E_i,j},l}, E_i)\mathbf{x}_{t_{\mathcal{T}_{E_i,j},l}}\|} \quad (4)$$

However, updating κ is not straightforward. We follow the first order approximation by Sra [2] and update κ as follows:

$$\kappa_a = \frac{\bar{R}(2-\bar{R}^2)}{1-\bar{R}^2} \quad (5)$$

$$\text{where, } \bar{R} = \frac{\|\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{l=1}^{k} p(z_a^l|t_{\mathcal{T}_{E_i,j},l}, E_i)\mathbf{x}_{t_{\mathcal{T}_{E_i,j},l}}\|}{\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{l=1}^{k} p(z_a^l|t_{\mathcal{T}_{E_i,j},l}, E_i)} \quad (6)$$

A. Jain, D. Das, J. K. Gupta and A. Saxena are with the Department of Computer Science, Cornell University, USA. ashesh@cs.cornell.edu, dd367@cornell.edu, jkg76@cornell.edu asaxena@cs.cornell.edu

[1]We extract the information about the environment and activities by querying OpenRAVE. In practice and in the robotic experiments, human activity information can be obtained using the software package by Koppula et al. [1].

Estimating Beta distribution parameters: Beta distribution is parameterized by two scalars α and β. We use method of moments to estimate these parameters. For an activity a, we first estimate first and second order moments i.e. sample mean and variance:

$$m_a = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{l=1}^{k} p(z_a^l|t_{\mathcal{T}_{E_i},j,l}, E_i)\bar{d}_{t_{\mathcal{T}_{E_i},j,l}}}{\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{l=1}^{k} p(z_a^l|t_{\mathcal{T}_{E_i},j,l}, E_i)} \quad (7)$$

$$v_a = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{l=1}^{k} p(z_a^l|t_{\mathcal{T}_{E_i},j,l}, E_i)(\bar{d}_{t_{\mathcal{T}_{E_i},j,l}} - m_a)^2}{\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{l=1}^{k} p(z_a^l|t_{\mathcal{T}_{E_i},j,l}, E_i)} \quad (8)$$

We then estimate α and β using the first and second order moments of data:

$$\alpha_a = m_a\left(\frac{m_a(1-m_a)}{v_a} - 1\right) \quad (9)$$

$$\beta_a = (1-m_a)\left(\frac{m_a(1-m_a)}{v_a} - 1\right) \quad (10)$$

Estimating Gaussian distribution parameters: It is parameterized by a scalar mean g and variance σ. For an activity a we estimate parameters of Gaussian distribution in closed form.

$$g_a = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{l=1}^{k} p(z_a^l|t_{\mathcal{T}_{E_i},j,l}, E_i)d_{t_{\mathcal{T}_{E_i},j,l}}}{\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{l=1}^{k} p(z_a^l|t_{\mathcal{T}_{E_i},j,l}, E_i)} \quad (11)$$

$$\sigma_a = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{l=1}^{k} p(z_a^l|t_{\mathcal{T}_{E_i},j,l}, E_i)(d_{t_{\mathcal{T}_{E_i},j,l}} - g_a)^2}{\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{l=1}^{k} p(z_a^l|t_{\mathcal{T}_{E_i},j,l}, E_i)} \quad (12)$$

In above equations $d_{t_{\mathcal{T}_{E_i},j,l}}$ is the distance of waypoint $t_{\mathcal{T}_{E_i},j,l}$ from object/human.

II. Application to Manipulation Scenarios

Unlike the 2-D navigation problem, manipulative tasks need to model the 3-D nature of the world and this problem depends on the object in hand and the objects encountered while following a trajectory. For example, bringing a knife close to any soft and fragile object is undesired. We therefore take an object centric view to ground each object pair interaction to a spatial distribution signifying object's functionality. Hence the total cost definition at each waypoint is modified as:

$$\mathbf{\Psi}(\mathcal{T} = \{t_1, ..., t_n\})|E = \prod_j\prod_i\prod_k \Psi_{a_j,a_k}(t_i|E) \quad (13)$$

Here a_j and a_k are attributes of the grabbed object and objects in the vicinity of waypoint t_i. These attributes are labels conveying physical properties of the objects. For example, a knife can have an attribute $sharp$, while a laptop can have attributes $electronic$ and $fragile$. These attributes are defined similar to Jain et al. [3].

Fig. 3: Relative Angle of Knife w.r.t sitting humans

A. Extending the Generative Model

We now extend the `PlanIt` generative model to object-object attribute pair interaction and learn the 3-D spatial distributions. For each attribute pair, we define a different cost function Ψ_{a_j,a_k}. An environment can have multiple instances of objects with same attributes and our overall cost function would have a cumulative effect of different attribute pairs formed, while moving through the waypoints.

1) Modified Affordance Representation:: The affordance representation gets slightly modified according to the grabbed object attributes. For example for a knife:

$$\Psi_{a_j,a_k}(t_i|E) = \begin{cases} \Psi_{a_j,a_k,dist}\Psi_{a_j,a_k,hei}\Psi_{a_j,a_k,ang} & \text{if } a_k \in \text{human} \\ \Psi_{a_j,a_k,dist}\Psi_{a_j,a_k,ang} & \text{if } a_k \notin \text{human} \end{cases} \quad (14)$$

Distance Preference $\Psi_{a_j,a_k,dist}(.)$: Humans would not prefer objects like knives to get very close to them or any fragile object i.e. the preferences vary with distance. This preference is captured in a 1-D Gaussian distribution centered around the object or human in the environment, parameterized by a mean and variance.

Angular Preference $\Psi_{a_j,a_k,ang}(.)$: Certain angular positions of the grabbed object w.r.t the human or object in the environment would be considered uncomfortable. For example, humans would not prefer knife pointed towards them, even if it is a reasonably far distance. This preference is captured by a *von-Mises* distribution as:

$$\Psi_{a_j,a_k,ang}(.) = \frac{1}{2\pi I_o(\kappa)} e^{\kappa\mu^T \boldsymbol{x}_{t_i}}$$

In the above equation, μ and κ are parameters that will be learned from the data, and x_{t_i} is a two dimensional unit vector representing the x and y projection of the angle between the grabbed-object orientation w.r.t to the object in the environment, where the coordinate system is defined locally for the attribute pair interaction.

Height Preference $\Psi_{a_j,a_k,hei}(.)$: It would not be preferable to move a *sharp* knife over delicate objects or humans. These preferences are captured by a beta distribution defined as:

$$\Psi_{a_j,a_k,hei}(.) = \frac{\bar{h}_{t_i}^{\alpha-1}(1-\bar{h}_{t_i}^{\beta-1})}{\boldsymbol{B}(\alpha,\beta)}; \bar{h}_{t_i} \in [0,1] \quad (15)$$

In the above equation, $\bar{h_{t_i}}$ is defined as:

$$\bar{h}_{t_i} = \begin{cases} \frac{h_{t_i}}{h_{obj}} & \text{if} \quad h_{t_i} < h_{obj} \\ \frac{h_{max}-h_{t_i}}{h_{max}} & \text{if} \quad h_{t_i} > h_{obj} \end{cases} \tag{16}$$

We learn the values of parameters α and β.

2) Parameter Learning: We optimize the data likelihood:

$$\Theta^* = \arg\max_{\Theta} \prod_{x=1}^{n} \prod_{y=1}^{m} \boldsymbol{\Psi}(\mathcal{T}_{E_x,y}|E_x;\Theta) \tag{17}$$

$$= \arg\max_{\Theta} \prod_{x=1}^{n} \prod_{y=1}^{m} \prod_{l=1}^{o} \sum_{k=1}^{|obj_{t_{\mathcal{T}_{E_x,y},l}}|} p(z_{a_j,a_k}|E_x;\Theta) \Psi_{a_j,a_k}(t_{\mathcal{T}_{E_x,y},l}|E_x;\Theta) \tag{18}$$

We use the Expectation-Maximization (EM) approach to learn the parameters. In the E-step, we calculate the posterior attribute pair assignment $p(z_{a_j,a_k}|t_{\mathcal{T}_{E_x,y},l}, E_x)$ for every waypoint and use this to update the parameters in the M-step.

E-step: Keeping the model parameters fixed we find the posterior probability of an attribute pair a_j, a_k at waypoint t:

$$p(z_{a_j,a_k}|t,E;\Theta) = \frac{p(z_{a_j,a_k}|E;\Theta)\Psi_{a_j,a_k}(t|E;\Theta)}{\sum_{h=1}^{|obj|} p(z_{a_j,a_h})\Psi_{a_j,a_h}(t|E;\Theta)} \tag{19}$$

M-step: Using the posterior from E-step we update the model parameters. Our affordance representation consists of three distributions: Gaussian, von-Mises and Beta. Gaussian parameters – mean (g) and variance (σ) and von-Mises mean (μ) can be updated in a closed form. We use Sra's [2] first order approximation to update von-Mises variance (κ). We use a similar approximation to update the beta distribution parameters (α and β) using the first and second order moments of the data.

Estimating Gaussian parameters: For an attribute pair a_j, a_k:

$$g_{a_j,a_k} = \frac{\sum_{x=1}^{n}\sum_{y=1}^{m}\sum_{l=1}^{o} p(z_{a_j,a_k}|t_{\mathcal{T}_{E_x,y},l}, E_x) d_{t_{\mathcal{T}_{E_x,y},l}}}{\sum_{x=1}^{n}\sum_{y=1}^{m}\sum_{l=1}^{o} p(z_{a_j,a_k}|t_{\mathcal{T}_{E_x,y},l}, E_x)} \tag{20}$$

$$\sigma_{a_j,a_k} = \frac{\sum_{x=1}^{n}\sum_{y=1}^{m}\sum_{l=1}^{o} p(z_{a_j,a_k}|t_{\mathcal{T}_{E_x,y},l}, E_x) (d_{t_{\mathcal{T}_{E_x,y},l}} - g_{a_j,a_k})^2}{\sum_{x=1}^{n}\sum_{y=1}^{m}\sum_{l=1}^{o} p(z_{a_j,a_k}|t_{\mathcal{T}_{E_x,y},l}, E_x)} \tag{21}$$

Estimating Beta distribution parameters: For an attribute pair a_j, a_k:

$$m_{a_j,a_k} = \frac{\sum_{x=1}^{n}\sum_{y=1}^{m}\sum_{l=1}^{o} p(z_{a_j,a_k}|t_{\mathcal{T}_{E_x,y},l}, E_x) \bar{h}_{t_{\mathcal{T}_{E_x,y},l}}}{\sum_{x=1}^{n}\sum_{y=1}^{m}\sum_{l=1}^{o} p(z_{a_j,a_k}|t_{\mathcal{T}_{E_x,y},l}, E_x)} \tag{22}$$

$$v_{a_j,a_k} = \frac{\sum_{x=1}^{n}\sum_{y=1}^{m}\sum_{l=1}^{o} p(z_{a_j,a_k}|t_{\mathcal{T}_{E_x,y},l}, E_x) (\bar{h}_{t_{\mathcal{T}_{E_x,y},l}} - g_{a_j,a_k})^2}{\sum_{x=1}^{n}\sum_{y=1}^{m}\sum_{l=1}^{o} p(z_{a_j,a_k}|t_{\mathcal{T}_{E_x,y},l}, E_x)} \tag{23}$$

We now use these to estimate α and β:

$$\alpha_{a_j,a_k} = m_{a_j,a_k} \left(\frac{m_{a_j,a_k}(1 - m_{a_j,a_k})}{v_{a_j,a_k}} - 1 \right) \tag{24}$$

$$\beta_{a_j,a_k} = (1 - m_{a_j,a_k}) \left(\frac{m_{a_j,a_k}(1 - m_{a_j,a_k})}{v_{a_j,a_k}} - 1 \right) \tag{25}$$

Estimating von-Mises distribution parameters: For an attribute pair a_j, a_k:

$$\mu_{a_j,a_k} = \frac{\sum_{x=1}^{n}\sum_{y=1}^{m}\sum_{l=1}^{o} p(z_{a_j,a_k}|t_{\mathcal{T}_{E_x,y},l}, E_x)\boldsymbol{x}_{\mathcal{T}_{E_x,y},l}}{||\sum_{x=1}^{n}\sum_{y=1}^{m}\sum_{l=1}^{o} p(z_{a_j,a_k}|t_{\mathcal{T}_{E_x,y},l}, E_x)\boldsymbol{x}_{\mathcal{T}_{E_x,y},l}||} \tag{26}$$

To update κ:

$$\kappa_{a_j,a_k} = \frac{\bar{R}(2 - \bar{R}^2)}{1 - \bar{R}^2} \tag{27}$$

where, $\bar{R} = \frac{||\sum_{x=1}^{n}\sum_{y=1}^{m}\sum_{l=1}^{o} p(z_{a_j,a_k}|t_{\mathcal{T}_{E_x,y},l}, E_x)\boldsymbol{x}_{\mathcal{T}_{E_x,y},l}||}{\sum_{x=1}^{n}\sum_{y=1}^{m}\sum_{l=1}^{o} p(z_{a_j,a_k}|t_{\mathcal{T}_{E_x,y},l}, E_x)}$

Estimating hidden variable: For an attribute pair a_j, a_k:

$$p(z_{a_j,a_k}|E;\Theta) = \frac{\sum_{x=1}^{n}\sum_{y=1}^{m}\sum_{l=1}^{o} p(z_{a_j,a_k}|t_{\mathcal{T}_{E_x,y},l}, E_x)}{N} \tag{28}$$

where, $N = m \times n \times o$

REFERENCES

[1] H. Koppula, R. Gupta, and A. Saxena, "Learning human activities and object affordances from rgb-d videos," *IJRR*, vol. 32, no. 8, 2013.

[2] S. Sra, "A short note on parameter approximation for von mises-fisher distributions: and a fast implementation of i s (x)," *Computational Statistics*, vol. 27, no. 1, 2012.

[3] A. Jain, B. Wojcik, T. Joachims, and A. Saxena, "Learning trajectory preferences for manipulators via iterative improvement," in *NIPS*, 2013.