Modeling Details of the Activity Tracker

(or how I learned to stop worrying and love the DBN)

Don Patterson

June 4, 2004

1 The Activity Graph

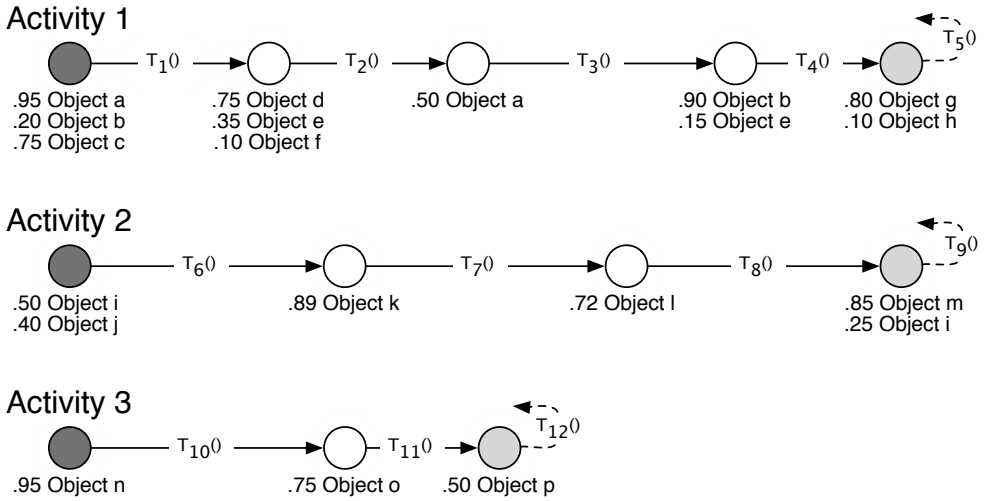

Figure 1: An example Activity Graph.

An *Activity Graph* (e.g., figure 1) is a graphical representation of the *Activity Space*. It shows how activities are structured, how observations effect the belief state and exhaustively represents the activities that the Activity Tracker believes that a user may be undertaking. It is not strictly a graphical representation of the state transitions in the system, but it does give an intuitive sense of how a user can move through the state space and it is sufficient to construct a complete state transition diagram, although such a diagram would be large and marginally useful if actually constructed.

The Activity Graph in figure 1 is interpreted as follows: there are three activities. The first activity consists of five steps. Each *Activity Node*, A, has a temporal distribution associated with each of its out-edges which is described by the function $T_x()$. The first activity, therefore, has a total time distribution of $\sum_{1:5} T_i()$. The second activity has four steps and the third activity has three

steps. Each activity has a node shaded dark gray which indicates the first step of the activity. The collection of dark gray shaded nodes form a set, $\hat{A}_b$.

There is an implicit edge from the end of every activity to the first node of every activity. This is represented in the figure by a dashed curved line originating from a light gray shaded node. The set of light gray shaded nodes which are the end of each activity also form a set, $\hat{A}_e$.

Each Activity Node in the graph also has a collection of objects associated with it. For a given node, A, the set of objects associated with that node is $\hat{O}_A$. These objects are the objects that are probabilistically expected to be touched at least once by a user while at the indicated node before proceeding to the next node in the activity. Each object is labeled with a probability of being touched. The probabilities are independent of each other and are combined using a Naive Bayes approximation. For example, while a user is in the first node of the first activity the probability of completing this step of the activity after touching objects a, b, and c is $P(\{a, b, c\} \mid \hat{O}_A) = (.95)(.20)(.75) = .1425$. The probability of completing this node by touching just objects a and b is $P(\{a, b, \neg c\} \mid \hat{O}_A) = (.95)(.20)(1 - .75) = .0475$.

We address sensor errors later in this paper, but in their absence we make the assumption that our model is correct. We don't explicitly account for execution errors. Furthermore, our model assumes that a user is engaged in a single activity at any given time, that the user completes that activity before beginning a new activity and that all activities are strictly linear orderings of Activity Nodes (although the structure of the Activity Node itself allows for some variation in ordering of the object touches "internally").

2 The Observables

An observation, $\hat{z}$, consists of a set of RFID tags. Observations are assumed to arrive at a regular rate and may indicate that no tags have been touched. Each RFID tag has a one-to-one mapping with objects in the Activity Graph, so an observation might be:

$$\hat{z} = \{Object\ a, Object\ b, Object\ c\}$$

A history of four observations might look like:

$$\hat{z}_{1:4} = \{\{Object\ a, Object\ b, Object\ c\}, \{Object\ d\}, \emptyset, \{Object\ e, Object\ f\}\}$$

3 Activity Graph as a Dynamic Bayes Net (DBN)

Before the Activity Tracker infers the current state of the system, the Activity Graph must be compiled into a graphical model. We use a 2TBN graphical model, a subset of Dynamic Bayes Nets which themselves are a type of Hidden Markov Model. Figure 2 shows the structure of our

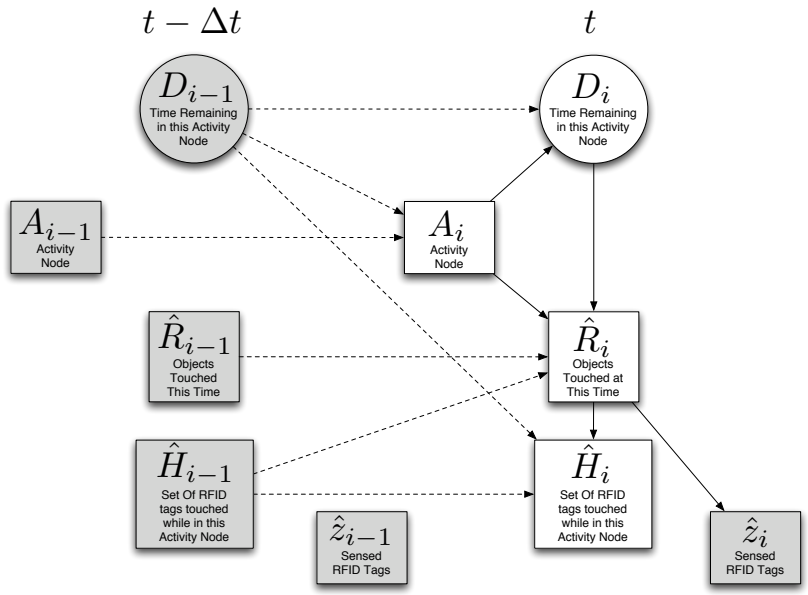

Figure 2: A Dynamic Bayes Net that describes the state transitions. Round nodes are continuous random variables. Square nodes are multinomial random variables. Shaded nodes are observed variables. Solid arrows are intra-temporal dependencies. Dashed arrows are inter-temporal dependencies. Time progresses from left to right.

model. As is standard in 2TBNs, when an observation is made, the state of the system at the previous time-step is frozen and treated as observed and the most likely state is computed over the hidden nodes in the current time-step.

The semantics of the 2TBN clearly show how an Activity Graph can generate the appropriate graphical model structure: the left half of the model represents the state of the system at the last observation. The previous state is treated as observed. The right half of the model is the state of the system at the current time step. Only the sensed RFID tags are observed in the current observation ($\hat{z}_i$). We assume that observations arrive at discrete regular intervals and that the state of the system does not change between observations.

3.1 Node Semantics

The ith state of the system is therefore, $x_i = \left\langle A_i, D_i, \hat{H}_i, \hat{R}_i, \hat{z}_i, \right\rangle$, and consists of the following variables:

- A is a hidden multinomial variable which has one value for every Activity Node in the Activity Graph. It represents the Activity Node that the user is executing.
- D is a hidden continuous variable which represents the amount of time remaining in the current Activity Node before switching to the next activity node.

- $\hat{H}$ represents the cumulative history of tags actually touched while executing the current Activity Node. It is also a multinomial variable which has one value for every member of the power-set of tags that are in the Activity Graph. This variable only "remembers" whether or not a tag has been touched, not how many times it has been touched.
- $\hat{R}$ is similar, but represents only the set of tags presently being touched. It is a hidden variable and is different than $\hat{z}_i$ because of sensor error in the system which might cause the RFID tags which are seen to be different than the tags which are actually touched. It is also a multinomial variable which has one value for every member of the power-set of tags that are in the Activity Graph.
- $\hat{z}$ represents the set of RFID tags sensed. It is an observed multinomial variable which has one value for every member of the power-set of tags that are in the Activity Graph.

3.2 Conditional Probability Semantics

We wish to determine the probability of being in a state given a sequence of observations. Using the Markov assumption [1] we can describe the probability of being in a state recursively as follows:

$$P(x_t \,|\, \hat{z}_{1:t}) = \int P(x_t \,|\, \hat{z}_t, x_{t-1}) \overbrace{P(x_{t-1} \,|\, \hat{z}_{1:t-1})}^{Prior} dx_{t-1} \tag{1}$$

$$P(x_1 \,|\, \hat{z}_{1:1}) = \int P(x_1 \,|\, \hat{z}_1, x_0) \overbrace{P(x_0)}^{Prior} dx_0 \tag{2}$$

3.2.1 Prior

The prior probability in Equation 1 is assumed to be known based on a recursive calculation. The base case, which is the probability expressed in Equation 2 is the a priori belief that the user is first observed in a given state.

In our application we assume that the first observation coincides with the user beginning an activity sequence in the Activity Graph (i.e, executing an Activity Node in $\hat{A}_b$). Among this subset of Activity Nodes we have a uniform prior belief that the user is starting an activity. This corresponds to a subset of the state space in the 2TBN. We can formally describe the prior belief in terms of the full state space as follows:

[1] Our use of the variable $\hat{H}$ allows us to capture the history of RFID tags observed without requiring multiple time step dependencies

$$P(x_0 = \left\langle A_{x_0}, D_{x_0}, \hat{H}_{x_0}, \hat{R}_{x_0}, \hat{z}_{x_0} \right\rangle) = \begin{cases} if & A_{x_0} \notin \hat{A}_b & \Rightarrow & 0 \\ else\ if & \hat{H}_{x_0} \neq \emptyset & \Rightarrow & 0 \\ else\ if & \hat{R}_{x_0} \neq \emptyset & \Rightarrow & 0 \\ else\ if & \hat{z}_{x_0} \neq \emptyset & \Rightarrow & 0 \\ else & & \Rightarrow & \frac{1}{|\hat{A}_b|} T_{A_{x_0}}(D_{x_0}) \end{cases} \tag{3}$$

Informally, this enforces that the model begins with no observed tags, an empty tag history and a uniform distribution over which activity the user is starting, but within that activity node, the belief is distributed according to the temporal distribution function, T_{A_x}.

The time distribution for each activity node is assumed to be normalized and therefore has the following property:

$$\forall A_x : \sum_D P(D|A_x) = \sum_D T_{A_x}(D) = 1 \tag{4}$$

The total probability of all initial states is shown to sum to 1 below:

$$\sum_r \sum_h \sum_a \sum_d P(x_0 \,|\, \hat{R}_x = r, \hat{H}_x = h, A_x = a, D_x = d) \;=\; \tag{5}$$

$$\sum_{r=\emptyset} \sum_{h=\emptyset} \sum_{a \in \hat{A}_b} \sum_d \frac{1}{|\hat{A}_b|} P(d \,|\, A_x) \;=\; \tag{6}$$

$$\sum_{r=\emptyset} \sum_{h=\emptyset} \sum_{a \in \hat{A}_b} \frac{1}{|\hat{A}_b|} \sum_d T_{A_x}(d) \;=\; \tag{7}$$

$$\sum_{r=\emptyset} \sum_{h=\emptyset} \sum_{a \in \hat{A}_b} \frac{1}{|\hat{A}_b|} \;=\; \tag{8}$$

$$\sum_{r=\emptyset} \sum_{h=\emptyset} 1 \;=\; 1 \tag{9}$$

3.3 Update Equation

By assuming independence between the current observation and the previous state given that the current state is unknown (the v-case of d-separation), we can further simplify equation 1 as follows:

$$P(x_t \,|\, \hat{z}_{1:t}) \;=\; \int P(x_t \,|\, \hat{z}_t) P(x_t \,|\, x_{t-1})\; P(x_{t-1} \,|\, \hat{z}_{1:t-1}) dx_{t-1} \tag{10}$$

$$\;=\; P(x_t \,|\, \hat{z}_t) \int P(x_t \,|\, x_{t-1})\; P(x_{t-1} \,|\, \hat{z}_{1:t-1}) dx_{t-1} \tag{11}$$

$$\;=\; \frac{P(\hat{z}_t \,|\, x_t) P(x_t)}{P(\hat{z}_t)} \int P(x_t \,|\, x_{t-1})\; P(x_{t-1} \,|\, \hat{z}_{1:t-1}) dx_{t-1} \tag{12}$$

We make the assumption of a stationary model in which for any given sequence of observations the prior distributions of x_t and $\hat{z}_t$ are constant. As such we can rewrite our equation as follows:

$$P(x_t \,|\, \hat{z}_{1:t}) \propto \underbrace{P(\hat{z}_t \,|\, x_t)}_{Sensor\ Model} \int \underbrace{P(x_t \,|\, x_{t-1})}_{State\ Transition\ Model} \underbrace{P(x_{t-1} \,|\, \hat{z}_{1:t-1})}_{Recursive\ Prior} dx_{t-1} \tag{13}$$

Each part of 13 has a specific interpretation with a counterpart in the DBN: The sensor model describes the probability of observing objects given sensor inaccuracies. The state transition model is derived from the Activity Graph and describes how a user moves through the space of Activity Nodes and the recursive prior is assumed to be known.

3.3.1 The Sensor Model Semantics

The sensor model which we use includes a model of RFID sensor measurement error, which we assume is independent of which tag we are reading. The false positive (FP) and false negative (FN) error rate for the RFID reader may be asymmetric:

$$\begin{aligned} & P(\hat{z}_t \,|\, x_t = \{A_{x_t}, D_{x_t}, \hat{R}_{x_t}, \hat{H}_{x_t}\}) && (14)\\ = \;& P(\hat{z}_t \,|\, \hat{R}_{x_t}) && (15)\\ = \;& \underbrace{(1 - P(FP))^{|\hat{z}_t \cap \hat{R}_{x_t}|}}_{True\ Positive} \underbrace{(1 - P(FN))^{|\neg \hat{z}_t \cap \neg \hat{R}_{x_t}|}}_{True\ Negative} \underbrace{P(FP)^{|\hat{z}_t - \hat{R}_{x_t}|}}_{False\ Positive} \underbrace{P(FN)^{|\hat{R}_{x_t} - \hat{z}_t|}}_{False\ Negative} && (16) \end{aligned}$$

The step from equation 14 to equation 15 is derived from the conditional independency relations in the 2TBN.

If there were no RFID sensor error, equation 16 would reduce to

$$P(\hat{z}_t \,|\, \hat{R}_{x_t}) = \begin{cases} if \quad \hat{z}_t = \hat{R}_{x_t} & \Rightarrow \; 1 \\ else & \Rightarrow \; 0 \end{cases} \tag{17}$$

3.3.2 The State Transition Model

The State Transition Model term in proportionality relation 13 reflects how our model assigns belief of a given state given the prior state. The update equation can be factored into several equations as Figure 2 suggests:

$$\begin{aligned} & P(x_t \,|\, x_{t-1}) \\ = \;& P(A_{x_t}, D_{x_t}, \hat{R}_{x_t}, \hat{H}_{x_t} \,|\, A_{x_{t-1}}, D_{x_{t-1}}, \hat{R}_{x_{t-1}}, \hat{H}_{x_{t-1}}) \\ = \;& \underbrace{P(A_{x_t} \,|\, A_{x_{t-1}}, D_{x_{t-1}})}_{Activity\ Progression} \underbrace{P(D_{x_t} \,|\, A_{x_t}, D_{x_{t-1}})}_{Timing} \underbrace{P(\hat{R}_{x_t} \,|\, A_{x_t}, D_{x_t}, R_{x_{t-1}}, H_{x_{t-1}})}_{Tag\ Expectation} \underbrace{P(\hat{H}_{x_t} \,|\, D_{x_{t-1}}, \hat{R}_{x_t}, \hat{H}_{x_{t-1}},)}_{Tag\ History\ Collection} \end{aligned}$$

Let's look at the semantics of each of these pieces in turn.

3.3.3 Activity Progression Semantics

Activity progression is defined by the Activity Graph. If the user is executing an activity node, when the timer becomes 0, the next node in the activity progression becomes active. The only exception occurs at the end of an activity in which case there is a uniform probability of moving to the beginning of any activity. In both cases the value of state variable A changes :

$$P(A_{x_t} \,|\, A_{x_{t-1}}, D_{x_{t-1}}) \;=\; \begin{cases} if & (D_{x_{t-1}} > 0) \wedge (A_{x_{t-1}} = A_{x_t}) & \Rightarrow & 1 \\ if & (D_{x_{t-1}} > 0) \wedge (A_{x_{t-1}} \neq A_{x_t}) & \Rightarrow & 0 \\ if & (D_{x_{t-1}} <= 0) \wedge (A_{x_{t-1}} \notin \hat{A}_e) \wedge (A_{x_{t-1}} \succ A_{x_t}) & \Rightarrow & 1 \\ if & (D_{x_{t-1}} <= 0) \wedge (A_{x_{t-1}} \notin \hat{A}_e) \wedge (A_{x_{t-1}} \not\succ A_{x_t}) & \Rightarrow & 0 \\ if & (D_{x_{t-1}} <= 0) \wedge (A_{x_{t-1}} \in \hat{A}_e) \wedge (A_{x_t} \in \hat{A}_b) & \Rightarrow & \frac{1}{|\hat{A}_b|} \\ if & (D_{x_{t-1}} <= 0) \wedge (A_{x_{t-1}} \in \hat{A}_e) \wedge (A_{x_t} \notin \hat{A}_b) & \Rightarrow & 0 \end{cases} \tag{18}$$

3.3.4 Timing Semantics

Timing is a straightforward counter which counts down deterministically to zero. The variable D_{x_t} keeps track of how much longer the user will be in the current activity node. When the previous counter is zero, the current counter is stochastically reset according to the temporal distribution associated with the current Activity Node.

It is updated as follows:

$$P(D_{x_t} \,|\, A_{x_t}, D_{x_{t-1}}) = \begin{cases} if & (D_{x_{t-1}} > 0) \wedge (D_{x_t} = D_{x_{t-1}} - \Delta t) & \Rightarrow 1 \\ if & (D_{x_{t-1}} > 0) \wedge (D_{x_t} \neq D_{x_{t-1}} - \Delta t) & \Rightarrow 0 \\ if & (D_{x_{t-1}} <= 0) \wedge (D_{x_t} = d) & \Rightarrow T_{A_{x_t}}(d) \end{cases} \tag{19}$$

3.3.5 Tag Expectation

Tag expectation refers to the expected belief of seeing a given set of tags.

$$P(\hat{R}_{x_t} \,|\, A_{x_t}, D_{x_t}, \hat{R}_{x_{t-1}}, \hat{H}_{x_{t-1}}) = \prod_{r \in \hat{R}_{x_t}} P'(r \,|\, A_{x_t}, D_{x_t}, \hat{R}_{x_{t-1}}, \hat{H}_{x_{t-1}}) \tag{20}$$

$$= \begin{cases} if & r \in \hat{R}_{x_{t-1}} & \Rightarrow & k_1 \\ if & r \notin \hat{R}_{x_{t-1}} \wedge r \notin \hat{O}_{A_{x_t}} & \Rightarrow & k_2 \\ if & r \notin \hat{R}_{x_{t-1}} \wedge r \in \hat{O}_{A_{x_t}} \wedge r \notin \hat{H}_{x_{t-1}} & \Rightarrow & \frac{P(r \,|\, \hat{O}_{A_{x_t}})}{D_{x_t}} \Delta t \\ if & r \notin \hat{R}_{x_{t-1}} \wedge r \in \hat{O}_{A_{x_t}} \wedge r \in \hat{H}_{x_{t-1}} & \Rightarrow & k_3 \end{cases} \tag{21}$$

The intuition for these dependencies is that it is calculated on a tag by tag basis (equation 20), and then the individual tag probabilities are multiplied to get the overall probability. k_1 is a large constant which is learned from the data which reflects the fact that a repeated observation from one time step to the next is the result of holding something and shouldn't bias one belief against another given that they both just saw a tag. k_2 is a small constant which reflects the likelihood of actually touching something which is not a part of the current activity. If you weren't allowed to touch anything else while performing an activity then this would be zero. k_3 is a constant learned from data which reflects the fact that once you see a tag, seeing the tag again while in the same activity node shouldn't bias you any more or less against other activity nodes in which you have already seen that object.

3.3.6 Tag History Collection

The tag history is maintained though a simple deterministic update of the tags that have been seen and is reset when an activity node switch is observed.

$$P(\hat{H}_{x_t} \,|\, D_{x_{t-1}}, \hat{R}_{x_t}, \hat{H}_{x_{t-1}},) = \begin{cases} if & (D_{x_{t-1}} = 0) \wedge (\hat{H}_{x_{t-1}} = \emptyset) & \Rightarrow & 1 \\ if & (D_{x_{t-1}} = 0) \wedge (\hat{H}_{x_{t-1}} \neq \emptyset) & \Rightarrow & 0 \\ if & (D_{x_{t-1}} \neq 0) \wedge (\hat{H}_{x_t} = \hat{H}_{x_{t-1}} \cup \hat{R}_{x_t}) & \Rightarrow & 1 \\ if & (D_{x_{t-1}} \neq 0) \wedge (\hat{H}_{x_t} \neq \hat{H}_{x_{t-1}} \cup \hat{R}_{x_t}) & \Rightarrow & 0 \end{cases} \tag{22}$$

4 Conclusion

This paper has proposed a method of constructing the structure and probability tables of a Dynamic Bayes Net from an Activity Graph for the purpose of inferring activities from a stream of RFID sensors.