Human State Estimation Through Learning Over Common Sense Data

William Pentney, Ana-Maria Popescu, Shiaokai Wang, Jeff Bilmes, Henry Kautz
Department of Computer Science & Engineering
University of Washington
Box 352350
Seattle, WA 98105
{bill,amp,shiaokai,bilmes,kautz}@cs.washington.edu

Matthai Philipose
Intel Research Seattle
1100 NE 45th Street
6th Floor
Seattle, WA 98105
matthai.philipose@intel.com

Abstract

We seek to tackle the problem of human state recognition, in which sensor-based observations are used to reason about the state of the general human environment. Recent work [Pentney *et al.*, 2006] has shown promise in using large publicly available hand-contributed commonsense databases as joint models that can be used to interpret day-to-day object-use data. We discuss the development of a statistical model for reasoning over large amounts of commonsense information about human activity, and the use of Web-based information retrieval techniques to evaluate and enhance such information for more effective use. Additionally, we discuss how to improve the performance of our model through the use of learning techniques which can scale to the very large networks induced by this commonsense data. Finally, we present experiments to show how these techniques can be used to provide improved results in the prediction of everyday human state.

1 Introduction

Sensor-based methods for inferring the state of people and their environment have a variety of applications including elder care management [**?**], institutional workflow management and proactive computing [**?**]. A system that could tell whether an elderly person living alone has a cold, has taken medication or is depressed, for instance, could substantially reduce the financial burden of care. A key challenge in building such systems is the need for models that relate low-level sensor signals (e.g., vision) to high-level concepts (e.g., depression). The conventional approach to acquiring such models is to apply machine learning techniques to labeled sensor data, given a "structure" prior on the dependencies between variables. The structure itself is often provided by hand by application developers.

This method, however, becomes quite expensive when trying to track and make predictions regarding the tens of thousands of aspects of daily life. A system to reason over everyday activity on such a scale requires a "commonsense" encoding of daily life and the numerous relationships between everyday objects, actions, and concepts - e.g. the fact that people will eat when they are hungry, or that flipping the lightswitch will turn on the light. There are two particular challenges with respect to creating such a representation. First, encoding such information generally requires human effort for manually entering the many commonsense facts that such reasoning requires. Second, using known reasoning techniques over such information can be computationally quite expensive, and may scale poorly due to the quantity of information necessary to perform meaningful inference.

Fortunately, many efforts to accumulate everyday commonsense knowledge exist; well-known efforts such as Cyc [Lenat and Guha, 1990] and OpenMind/OMICS [Singh *et al.*, 2002; Gupta and Kochenderfer, 2004] have been devoted to accumulating and codifying this information. The OMICS database has aggregated many relational predicate groundings contributed by anonymous web surfers, providing information such as "An action associated with the object 'cereal' is 'eat"'. Since emerging *dense* sensor networks [Fishkin *et al.*, 2005] can directly report high-level object-use data, and since the OMICS database is grounded extensively in terms of object use, it is feasible to automatically obtain a commonsense interpretation of the world state by connecting dense sensors to a propositional model representing the OMICS database. Here we will describe a system that produces a statistical model for reasoning over many possible variables pertaining to the state of the world in a manner that provides both reasonable efficiency and reasonable accuracy.

2 Common Sense Representation

A diagram of the architecture of our system, called SRCS, may be seen in Figure **??**; this is an augmented version of the system described in [Pentney *et al.*, 2006]. SRCS takes as its input the OMICS commonsense database, which contains roughly 50,000 instances of roughly 15 relations on a small set of domains

describing day-to-day life. Domains include `Object` and `Action`, which denote physical objects and actions performed by people respectively. Relations include `people(Action,Context)`, which relates actions to their contexts; an instance of this relation may be `people(eat, hungry)`. SRCS transforms these relations in a sequence of steps into a set of weighted Horn clauses, in which each antecedent and consequent represents a random variable about the state of the world (e.g. `userInState(hungry)`, `stateOf(kitchen light,on)`). These random variables are encoded into a large probabilistical graphical model (PGM) called a chain graph [Buntine, 1995]. Some nodes of the chain graph represent the use of objects; by connecting these nodes to actual observations of the use of these objects, we may provide observations to use for inference over other propositions about the world (e.g. "The user is hungry".)

Although the random variables in SRCS's Horn clauses could represent predicate groundings in first-order logic, the reasoning is currently done in a propositional fashion for efficiency; we do not reason more generally over the truth of predicates.

While OMICS provides us with many valuable commonsense facts, it also suffers from semantic gaps and noisy information. Facts that could be represented amongst the random variables induced by the above process (e.g. *use(lightswitch)* $\Rightarrow$ *stateOf(kitchen light,on)*) may not be represented, and some facts may be irrelevant or simply nonsensical. We thus make use of Web mining techniques, in the form of the KnowItAll system [Etzioni *et al.*, 2004], both to evaluate the predicates OMICS provides and to mine new predicates that may produce additional Horn clauses. For example, KnowItAll successfully mines the fact *contextactions(toothpaste, squeeze, brushing teeth)*, whose semantic meaning is "You will squeeze the toothpaste when you are brushing your teeth". These supplemental facts help fill in gaps in the existing OMICS data.

3 Probabilistic Graphical Model

To reason about the environment, we view the state of the world as a collection of random variables, each representing the truth of a Boolean predicate, such as "The light is on" or "The user is making the bed", for an interval of time t. We wish to track these facts over a succession of such time intervals. There will be dependencies between related variables; for example, if "The user is making a sandwich" is true, this should imply, with some relatively high probability, the truth of the predicate "The user is hungry". In the following sections, we describe in more detail how this model is constructed.

3.1 PGM Construction

As described, our processing of the OMICS data produces approximately 60,000 random variables representing facts about the environment at a given moment in time, and defines potential functions upon these random variables. We represent each of these random variables

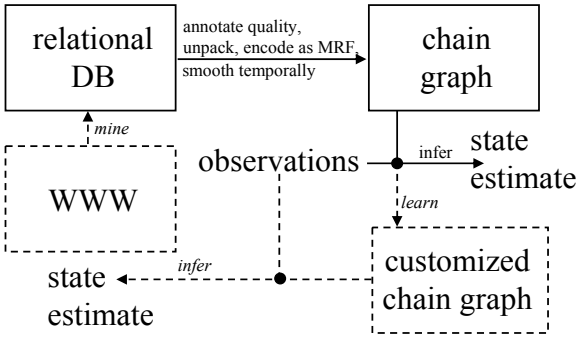

Figure 2: System architecture.

as nodes in a chain graph, a mixed directed/undirected graphical model. Let $\mathbf{p_t} = (\mathbf{p_{1_t}}, ..., \mathbf{p_{n_t}})$ be random variables representing the truth of all predicates defined by the process previously described at time t. Let $\mathbf{o_t} = (\mathbf{o_{1_t}} ... \mathbf{o_{r_t}})$ be Boolean random variables, each representing the use of object o_i at time t when true. The set of random variables $\mathbf{p_t}$, observations $\mathbf{o_t}$ and functions $\Phi = (\phi_1, \phi_2 ... \phi_M)$ can be used to define a graph G_t representing a conditional random field [Lafferty *et al.*, 2001] over $\mathbf{p_t}$. However, we also must consider the relationships of variables between time slices. We rely on the relatively simple assumption that, independent of other influence, true predicates at t will remain in the state they are in with some fixed probability σ. To incorporate temporal data, we can then use a "double-slice" graph in which the graphs G_t and G_{t+1} are connected by directed edges from each p_{i_t} to $p_{i_{t+1}}$. We may then define a set of temporal potential functions ψ_i in which $\psi_i(p_{i_t}, p_{i_{t+1}}) = \sigma_T$ if both predicates are true, σ_F if they are both false, $1 - \sigma_T$ if p_{i_t} is true and $p_{i_{t+1}}$ false, or $1 - \sigma_F$ otherwise. The probability of an assignment at time $t + 1$ is

$$\hat{P}_t = P(\hat{\mathbf{p}}_{t+1} | \hat{\mathbf{p}}_t) = \frac{1}{Z} e^{(\sum_i \mu_i \psi_i(\hat{p_{i_t}}, \hat{p}_{i_{t+1}}) + \sum_i \lambda_i \phi_i(\hat{\mathbf{p}}_t, \hat{\mathbf{o}}_t))}$$

where λ_i and μ_i are weights on features ϕ_i and ψ_i. This defines a chain graph over $\mathbf{p_1} ... \mathbf{p_N}$ for data over time slices $1..N$, as depicted in Figure 1. The model can also be seen as an HMM in which the full set of non-observation random variables at t represents the hidden state of t. We use the BP algorithm of Pearl [Pearl, 1988] to obtain a decent approximation of the distribution.

4 Learning

To improve the quality of inference, we employ machine learning techniques to improve SRCS's prediction. Given our model, we can optimize the log-likelihood of a set of training data D with respect to the vector of weights λ,μ, i.e. we wish to maximize $\log P(D|\lambda, \mu)$ over λ, μ. This calculation, however, is intractable given the size of our graph. We must thus employ approximation techniques. One option is to estimate $\log P(D|\lambda, \mu)$ using loopy BP over each time slice; however, even this can be computationally expensive over large amounts of data. Another method is to instead optimize the log of the pseudolikelihood $PL(D)$ [**?**], a more computationally efficient approximation to the likelihood used in some contexts for weight learning on MRFs [**?**]. The log pseudolikelihood of $\hat{\mathbf{p}}_t$ given $\hat{\mathbf{p}}_{t-1}$ is $\log PL(\hat{\mathbf{p}}_t|\hat{\mathbf{p}}_{t-1}) = \sum_{i=1..n} \log P(p_{i_t}|MB(p_{i_t}))$, where $MB(p_{i_t})$ is the Markov blanket of p_{i_t}. Given this objective function, we can maximize $\log PL(D)$ over possible λ, μ using standard convex optimization techniques. We employ the L-BFGS algorithm for this optimization, adding in a Gaussian prior as in [**?**] to prevent overfitting.

4.1 Thresholding

The probabilities that SRCS outputs, are currently not an effective as an *absolute* measure of a predicate's distribution, but are still often useful as a *relative* measure. To label object use traces, we feed SRCS traces labeled with ground truth for the variables being tracked, and train decision stumps on each proposition to recognize the optimal threshold value, as measured by a weighted f-measure, for labeling variables. We then perform inference over object traces via the technique described and label according to whether the probabilities output fall above or below the learned thresholds.

5 Clustering for Inference and Learning

While one can perform inference on a full double timeslice graph with BP to infer the probability of the state of the environment, it is computationally quite expensive this manner requires about 30 minutes on our system. In [Pentney *et al.*, 2006], this issue was resolved through pruning the graph to a subgraph G', defined by the union of all breadth-first traversals to a depth d on each fact p_i (in the paper, $d = 2$.) While this is an effective technique, it requires foreknowledge of what predicates one wishes to query over before constructing the graph and the input of observations for inference.

To improve efficiency in another manner, we take advantage of some intuition about the structure of our graph. Our collection of everday commonsense predicates contains many predicates whose respective states are likely to be mostly unrelated. Given this intuition, we seek to partition the timeslice graph into subgraphs based on the features defined upon the graph. We cluster the graph G, containing both predicates $\mathbf{p}$ and observations $\mathbf{o}$, into k distinct clusters $C_1, ...C_k$ using single-link clustering, with the weight of the edge between p_{i_t} and p_{j_t} defined by the weight λ_i of a potential ϕ_i dependent on both i and j, if it exists, and zero otherwise. Under this scheme, clusters with high potentials between their respective weights are merged in the iterations of the clustering algorithm. Intuitively, this corresponds to the discovery of subsets of predicates which have strong relationships with each other, and likely share a similar context (e.g. the predicate `location(kitchen)` is likely to be associated with kitchen implements).

The clustering gives us a means of approximating queries on specific nodes based on evidence. If $c(x)$ is the cluster of node x, then given that observed nodes $o_1, ...o_m$ and we wish to query over $q_1, ...q_s$, we may limit our inference to the subgraph represented by the union of all $c(q_i)$ and $c(o_i)$. Thus we are able to "zoom in" on the nodes whose value is most likely to be affected by the observations, or which nodes we are most interested in querying.

Our clustering technique can help improve the efficiency of learning as well; when performing inference during learning, we may approximate the likelihood of the training data with clustered inference in a similar manner, calculating $PL(D)$ strictly over subgraphs defined by the clustering described in the previous section. In our model, the majority of the predicates in labeled data are going to be false at any given time, and most of the predicates in the graph will, in fact, never be seen as true (some of them are not measurable, irrelevant, or even nonsensical). To minimize the number of terms we need to compute in our objective function, we consider $\mathbf{C}_t$, the union of the clusters of every node labeled true in the slice t, along with a sampling of false query nodes, and then use as our objective function $\log APL(D) = \sum_{t=1..N} \log APL(\hat{\mathbf{p}}_t) = \sum_{\mathbf{C}_t} \log P(p_{i_t}|MB(p_{i_t}))$. Intuitively, by using this objective function we are only considering terms relating to predicates that appear as both true and false, as well as predicates in their neighborhood. The sampling of false query nodes is included to ensure a sampling of false instances of each predicate that appears is also included in the learning of the objective function.

6 Experiments

For our experimental evaluation, we used traces of household object use in an experimental setting as produced by three users while performing various daily activities in a simulated home environment, as used in [Pentney *et al.*, 2006]. Data was collected via the use of an RFID reading bracelet in conjunction with tagged household objects. A total of 5-7 minutes worth of performance of each activity was collected, for a total of approximately 70-75 minutes of data. These traces were divided into time slices of 2.5 seconds; reasoning was to be performed over each of these time slices.

For these activities, we considered a variety of variables about the state of the world which could be relevant to these activities. We then selected a set of 24 Boolean variables in the collected SRCS database which repre-

Method	Learning	Acc	Prec	Recall
Random	-	50.00%	8.07%	50.00%
All false	-	93.00%	-	0.00%
Orig	No	81.81%	24.85%	51.34%
Orig	Yes	83.40%	23.80%	67.22%
Mined	No	84.92%	22.62%	52.11%
Orig/Mined	no	86.01%	30.20%	83.55%
Clustered	no	73.58%	16.43%	72.18%
Clustered	Clustered	79.94%	21.29%	74.32%

Figure 3: Results from SRCS experiments.

sented these variables, or were semantically very close to them, such as `stateof(cereal, prepared)`. We then recorded their "truth" value as being true or false for each interval of time in the trace.

In [Pentney *et al.*, 2006], we performed inference on the same set of variables on a system using the KnowItAll priors but no learning of weights; here we perform learning in the method described before performing inference. One way we judge the effectiveness of our classifier is simply by judging its accuracy in labeling facts true or false. However, in practice most predicates are false most of the time (approx. 94% of the labels in the test data are false), and we would be more interested in correct labeling of true predicates in practice. We thus also measure the precision and recall of the classifier with respect to labeling the true predicates.

First, we ran our system with the graph pruned to depth $d = 2$ from the query predicates, with and without learning, with the original SRCS weights and mined quality measures from [Pentney *et al.*, 2006]. Next, we ran our system with newly mined predicate groundings discovered using KnowItAll. We then ran the system with the combined data. Finally, we also ran the system using the clustered inference method, with and without the clustered method of learning, so as to test the effectiveness of a (presumably) less accurate but more efficient classifier.

Our results are shown in Figure 3; we compare the performance to baselines of both random selection and labelling all facts as "false". We see that using learning on the system with a relatively small amount of training data provides some increase in accuracy and a considerable improvement in recall. Additionally, we find that adding predicate groundings mined using KnowItAll improves all measures. The accuracy and precision of the algorithms using clustering is lower, but still provide good recall. Clustered inference also takes less time (about 50% less) to run than inference on the whole graph; the loss of accuracy may be acceptable in some circumstances due to the improved efficiency.

7 Conclusions and Future Work

We have presented SRCS, a model for human state estimation which makes use of available large-scale commonsense data, Web mining, and machine learning techniques to provide good prediction of the state of the everyday human environment using sensor data. We have shown that such a system provides good accuracy in prediction with little direct human effort in specifying the necessary commonsense data, and that machine learning techniques, used in conjunction with labeled data traces, may provide improved performance in such prediction. There remains much ground for future work in improving both the prediction accuracy and efficiency of such a system; we continue to investigate methods of improving the efficiency of inference using more advanced techniques of isolating relevant subgraphs, and more efficient means of performing learning over large amounts of labeled or partially labeled data.

References

[Buntine, 1995] W. Buntine. Chain graphs for learning. In *UAI 1995*, 1995.

[Etzioni *et al.*, 2004] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Methods for domain-independent information extraction from the web: An experimental comparison. In *AAAI*, pages 391–398, 2004.

[Fishkin *et al.*, 2005] Kenneth P. Fishkin, Matthai Philipose, and Adam Rea. Hands-on RFID: Wireless wearables for detecting use of objects. In *ISWC 2005*, pages 38–43, 2005.

[Gupta and Kochenderfer, 2004] Rakesh Gupta and Mykel J. Kochenderfer. Common sense data acquisition for indoor mobile robots. In *AAAI*, pages 605–610, 2004.

[Lafferty *et al.*, 2001] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabalistic models for segmenting and labeling sequence data. In *Proc. of 18th International Conference on Machine Learning (ICML)*, 2001.

[Lenat and Guha, 1990] D. Lenat and R. V. Guha. *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*. Addison-Wesley, 1990.

[Pearl, 1988] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kauffman, 1988.

[Pentney *et al.*, 2006] W. Pentney, A. Popescu, S. Wang, H. Kautz, and M.Philipose. Sensor-based understanding of daily life via large-scale use of common sense. In *Proceedings of AAAI*, 2006.

[Singh *et al.*, 2002] Push Singh, Thomas Lin, Erik T. Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. OpenMind: knowledge acquisition from the general public. In *Proceedings of the First International Conference on Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems*, 2002.