A context-sensitive and user-centric approach to developing personal assistants

Marie desJardins and **Eric Eaton**
University of Maryland Baltimore County
Computer Science and Electrical Engineering Department
1000 Hilltop Circle, Baltimore MD 21250
{mariedj|eeaton1}@cs.umbc.edu

Kiri Wagstaff
Jet Propulsion Laboratory
Machine Learning Systems Group
4800 Oak Grove Drive, Pasadena CA 91109
kiri.wagstaff@jpl.nasa.gov

Abstract

Several ongoing projects in the MAPLE (Multi-Agent Planning and LEarning) lab at UMBC and the Machine Learning Systems Group at JPL focus on problems that we view as central to the development of persistent agents. This position paper describes our current research in this area, focusing on four topics in particular: effective use of observational and active learning, utilizing repeated behavioral contexts, clustering with annotated constraints, and learning user preferences.

Introduction

The ultimate goal of our research is to develop interactive, knowledge-rich, persistent agents that can operate in complex environments over an extended period of time. Such agents should be able to adapt as the environment changes, to incorporate knowledge of various types from multiple sources, to learn incrementally, and to leverage the results of earlier experiences in order to improve their performance on later tasks. Dr. desJardins's Ph.D. dissertation presented a framework for autonomous, goal-directed learning in stochastic domains (desJardins 1992). She and her students at UMBC continue to work on related problems. Similarly, Dr. Wagstaff's dissertation introduced the idea of constrained clustering, which provides a means for more effective user interaction with clustering methods (Wagstaff 2002); she continues to work in this area and to explore other ways to leverage background knowledge in learning.

In this position paper, we describe four areas of research related to the development of persistent agents that we are currently pursuing: combining observational and active learning, exploiting repeated behavioral contexts, clustering with annotated constraints, and learning user preferences. These projects are all in their early stages.

In the rest of the paper, we illustrate each research focus within an e-mail sorting domain. In this application domain, a persistent agent acts as a personalized e-mail assistant for multiple users. The assistant receives and sends e-mails, files messages in the appropriate folders, prioritizes unread messages, and alerts users when important messages arrive.

Effective Use of Observational and Active Learning

The goal of this work is to develop learning methods for persistent agents that minimize the burden on the user of providing training instances. Specifically, we are interested in developing techniques to integrate observational learning with active learning.[1]

A personal agent can learn many tasks by observing the user's actions, in much the same manner as apprentices traditionally learned their trade. This sort of non-intrusive *observational learning* does not interfere with the user's activities, and is therefore unrestricted in its application.

Personal assistants can learn from both user actions and inactions. Learning from user actions is rather straightforward and intuitive. Many user inactions can be interpreted as acceptance (including via cooperation and coadaptation) of the system's behavior. For example, once the e-mail assistant generalizes several user corrections (the *action* of moving e-mails to their proper folders) to a global sorting rule and applies it to other e-mails, the assistant can interpret the lack of further correction as acceptance of the generalized rule. The system can also interpret the user navigating directly to an e-mail that was sorted by the global rule as acceptance of that rule for that message.

For many tasks, observational learning requires an unacceptably high number of training instances. Active learning alleviates this problem, but requires user interruption and interaction. Personal assistants that actively learn must balance the need for information with the cost of disrupting the user.

In the framework of multiple contexts, the system must additionally balance between interrupting the user to obtain information relevant only to an isolated context (thereby slowing down the completion of this one task), and actively obtaining information relevant to a repeated context (potentially giving this task a lower amortized time over all contexts).

[1] Note that we use the term "active learning" for all learning methods that query the user, going beyond the typical usage in machine learning of querying only about class membership of unlabeled instances.

Context Awareness

Personal assistants reside in online, dynamic environments. In such domains, as users switch tasks or change their focus of attention, the context of learning changes as well. As a result, the target concept to be learned may shift to another concept. These multiple contexts are interleaved in time, and each context may include one or more concepts that need to be learned. Of particular interest for our research is the case where contexts *reappear* after some period of time.

Consider an intelligent e-mail client that emphasizes the e-mails that are related to the user's current task. The user may work on the annual company budget for a few hours and then on an unrelated Project X before resuming work on the budget. In the absence of explicit notifications, the e-mail client must infer the context governing the emphasis of e-mails from the user's behavior. When the user resumes work on the budget, the e-mail client should immediately return to emphasizing the e-mails that it had previously learned were budget-related. These two contexts, "budget" and "Project X," each contain a concept governing e-mail emphasis, but the concepts differ.

In this model, the user does not provide explicit notifications of the current context; rather, the system must infer the context based on a temporal stream of perceptions. The absence of explicit notifications prevents user irritation from continually having to notify the system of the current context. Additionally, the user's perception of the current context might differ from the system's perception of the current context. For example, the naïve user might have separate contexts for "January Budget," "February Budget," and so on, even though the same set of concepts is appropriate for each. The system should also be able to dynamically add new contexts as necessary and delete (or archive) old contexts. In practice, a personal assistant should also have the option to actively detect (via queries) the current context.

As demonstrated by Widmer and Kubat (1996), detecting context changes is important, since previously learned concepts for one context might hinder the learning of the concepts for another context. Upon encountering a previously seen context, the personal assistant should be able to resume learning from where it left off. We are developing an ensemble-based framework to solve this problem in an online domain with limited perceptions per time step and without explicit context notifications. To the best of our knowledge, no system has yet been able to successfully achieve this in the absence of explicit context notifications. Additionally, as the number of observations per time step decreases, the difficulty of detecting a context change and identifying the new context increases. The ensemble framework will also permit concept drift to allow for changing user behavior within a context.

We view context awareness as a meta-learning problem: given a temporal data series of observations and query-response pairs, what is the appropriate set of concepts to learn and follow? A similar problem has already been studied in a batch framework by Weigend, Mangeas, and Srivastava (1995) using gated experts.

Annotated Constraints for Clustering

Recent work on constrained clustering (Wagstaff 2002; Bilenko, Basu, & Mooney 2004) has resulted in methods to incorporate background knowledge in the form of *same* and *not-same* constraints into clustering algorithms. We are extending these methods to include *feature relevance annotations* on these constraints. We hypothesize that leveraging this additional user-provided knowledge will improve clustering performance for a given number of constraints.

Constraints are a simple and natural method of interaction that have been shown to be a useful source for background knowledge. Feature relevance annotations, which indicate which features the user found most useful in determining whether a pair of instances belong to the same cluster or not, seem like a natural extension of this idea.

The simplest way to perform constrained clustering is to treat each constraint as an *instance-level* or pairwise constraint (Wagstaff 2002). In this approach, only clusters that respect the constraints are allowed, but none of the unconstrained instances are directly affected. A different approach is to interpret user constraints as *space-level* information, generalizing *same* and *not-same* constraints to nearby instances by warping the similarity metric. With space-level constraints, instances near either endpoint of a *same* constraint are pulled into the same cluster, while instances near either endpoint of a *not-same* constraint are pushed into different clusters. This approach has been shown to be an effective technique that can yield improved generalization performance in many cases (Klein, Kamvar, & Manning 2002).

We treat feature annotations on constraints analogously to space-level constraints by warping the similarity metric—but only along the dimensions corresponding to the features mentioned in the annotations. In other words, other nearby instance pairs will be more or less likely to be grouped together based on their similarity in the specified dimensions.

We have implemented this approach to annotated constrained clustering as an extension to the MPCK-means algorithm (Bilenko, Basu, & Mooney 2004). Our initial experimental results justify our hypothesis that leveraging this additional user-provided knowledge will result in improved clustering performance for a given number of constraints over standard unconstrained clustering and several constrained clustering methods (Wagstaff 2002; Bilenko, Basu, & Mooney 2004). We are also exploring two alternative approaches that treat the annotations more globally, by using them to weight the specified features more heavily in the similarity metric either for the entire clustering problem, or on a per-cluster basis.

We believe that this work will lead to more natural ways to interact with the user. For example, in the e-mail domain, the assistant could use constrained clustering to form subgroups for "cluttered" mail boxes, with minimal guidance from the user (in the form of constraints) about which messages belong together, which messages should be filed separately, and why. We also hope that learned feature relevance knowledge, and the resulting similarity metrics, can be applied by a persistent agent to other clustering or classification problems in similar domains.

User Preferences

Representing, eliciting, and learning preferences have become active areas of research over the last several years (Cohen, Schapire, & Singer 1999; Boutilier, Bacchus, & Brafman 2001). However, there are still many open problems. In the machine learning community, the goal has generally been to find a pairwise ordering scheme that is maximally consistent with the user's preferences, as given through a training set of labeled (ranked) instances. This pairwise ordering is then used to construct a complete ranking of a set of unlabeled instances. Decision theory and preference elicitation has focused on modeling the tradeoffs between alternatives, and on constructing a preference model with the fewest queries possible.

In the above work, the focus is on generating a single ordering or ranking over a set of objects. By contrast, researchers working on methods to support combinatorial auctions in electronic marketplaces have developed bidding languages that can represent the utility of a *group* of objects (Nisan 2000). These languages have the limitation that they are purely propositional: the set of objects must be known in advance, and bids are on specific object sets.

We are interested in a related problem, of preferences over sets of attributed objects. That is, if we have an arbitrary collection of items that can be described in terms of a set of features, and want to present the user with the k "best" items, which items should we show them? The answer may not be "the k top-ranked items." For example, search engines typically do *not* show the k top results; rather, they do some form of clustering to group very similar results, and show representative items from the k top clusters.

We are currently developing a first-order language for representing preferences over sets. This language will allow us to specify domain- or context-specific requirements for a result set. The next step will be to develop algorithms that can be used to learn these set-level preferences directly from labeled user data, actively by generating queries, and/or observationally by inferring the user's reaction to a set of displayed items, based on the actions that they take.

Ultimately, our goal is to apply context-sensitive learning methods, as outlined previously, to learn task- and situation-specific preference rules that apply at different times. In our illustrative e-mail domain, when the user arrives in the morning and requests a summary of the most important new messages, the personal assistant might place a selection of high-priority messages on a variety of topics near the top of the message listing (as opposed to k very highly ranked messages that all refer to the same project). However, if the user is working on a particular project, the agent would give top priority to messages relevant to that project. As another example, if the user is very time-constrained, the most urgent messages should be displayed prominently; if the user plans a longer work session, so that more messages will be processed in total, then messages can be prioritized and grouped by topic rather than by urgency.

Conclusion

The ideas presented in this paper are user-centric, focusing on effective user interaction and on responsiveness to the user's changing needs. Our discussion has focused on a personal agent for a single user, but we could also develop persistent assistants for a *community* of users. Although a personal assistant must adapt to an individual user, there are many commonalities across users that agents could learn and incorporate into their behavior. Persistent agents that assist multiple users could generalize and transfer preferences and context-based behaviors across these users. On a larger scale, multiple personal assistants could communicate with each other in order to learn from the experience of other agents. We believe that we have identified some key issues that will move us closer to the ultimate goal of developing persistent assistants.

Acknowledgements

Eric Eaton is supported by a fellowship from the Goddard Earth Sciences and Technology (GEST) Center at UMBC. This research is also supported by NSF ITR grant #0325329. We thank Craig Cambias, Natalie Podrazik, and Qianjun Xu for their contributions to the ideas presented here.

References

Bilenko, M.; Basu, S.; and Mooney, R. J. 2004. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the International Conference on Machine Learning (ICML-2004)*, 81–88. ACM Press.

Boutilier, C.; Bacchus, F.; and Brafman, R. I. 2001. UCP-Networks: A directed graphical representation of conditional utilities. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence (UAI-2001)*, 56–64. Morgan Kaufmann.

Cohen, W. W.; Schapire, R. E.; and Singer, Y. 1999. Learning to order things. *Journal of AI Research* 10:243–270.

desJardins, M. 1992. *PAGODA: A Model for Autonomous Learning in Probabilistic Domains*. Ph.D. Dissertation, UC Berkeley. (Available as UCB CS Dept. Technical Report 92/678).

Klein, D.; Kamvar, S. D.; and Manning, C. D. 2002. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *Proceedings of the International Conference on Machine Learning (ICML-2002)*, 307–314. Morgan Kauffman.

Nisan, N. 2000. Bidding and allocation in combinatorial auctions. In *Proceedings of the Second ACM Conference on Electronic Commerce (EC-2000)*, 1–12. ACM Press.

Wagstaff, K. 2002. *Intelligent Clustering with Instance-Level Constraints*. Ph.D. Dissertation, Cornell University.

Weigend, A. S.; Mangeas, M.; and Srivastava, A. N. 1995. Nonlinear gated experts for time-series: Discovering regimes and avoiding overfitting. *International Journal of Neural Systems* 6(4):373–399.

Widmer, G., and Kubat, M. 1996. Learning in the presence of concept drift and hidden contexts. *Machine Learning* 23(1):69–101.