Latent Goal Analysis for Dimension Reduction in Reinforcement Learning

Matthias Rolf and **Minoru Asada**
Osaka University, Dep. Adaptive Machine Systems

1 Introduction

Reinforcement learning is a paradigm that is both very general and widely applied for interacting agents. Despite tremendous progress on both model-based and model-free algorithms, reinforcement learning does however still requires a substantial amount of manual task design. One of the major burdens for a truly autonomous operation of RL agents is the design of task-appropriate features [Kober and Peters, 2012] of state and action. These features need to be comprehensive for RL to perform effectively, yet compact in terms of dimension to perform efficiently.

In particular dimension reduction in reinforcement learning is a tedious issue. While standard dimension reduction of unsupervised problems (e.g. PCA) has to deal only with a single and fixed probability distribution, reinforcement has a distribution of environmental states, and a space of actions (with no adhoc probability distribution), and a scalar reward. Unsupervised schemes can be employed in RL to reduce the dimension of the environmental state [Legenstein et al., 2010] or believe state [Roy and Gordon, 2002, Poupart and Boutilier, 2002], but such schemes can not account for the actions (except when expert demonstrations are given [Bitzer, 2011]). Another attempt has been to learn reduced rank regression of transition probabilities of state *and* action to guide exploration [Nouri and Littman, 2010], but which cannot account for the actual relevance with respect to the reward. The only existing models that can consider states, actions, and reward at the same time estimate the reward function based on bi-linear regression and reduce the rank of the parameter matrix [Koren et al., 2009, Chu and Park, 2009].

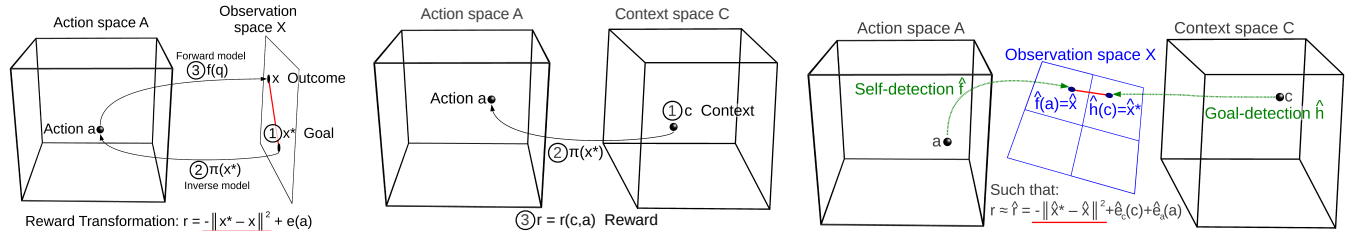
In contrast to reinforcement learning, adaptive control formulations [Nguyen-Tuong and Peters, 2011] already come with expressive and typically low-dimensional goal and task representations, which have been generally considered more expressive than the RL setting [Kaelbling et al., 1996]. Goal and actual values in motor control define a relation similar [Rolf and Steil, 2014] to actual and target outputs in classical supervised learning settings by providing "directional information" in contrast to a mere "magnitude of an error" in reinforcement learning [Barto, 1994]. Recent work [Rolf and Asada, 2014] however showed that these two problem formulations can be transformed into each other. Hence, highly descriptive task representations can be extracted out of reinforcement learning problems by transforming them into adaptive control problems. After introducing the method called Latent Goal Analysis, we discuss the possible application of this approach as dimension reduction technique in reinforcement learning. Experimental results in a web recommender scenario confirm the potential of this technique.

2 Latent Goal Analysis

In order to derive a mathematical learning rule, we first introduce the basic formalism of adaptive control or coordination problems [Chung et al., 2007, Nguyen-Tuong and Peters, 2011, Rolf et al., 2010] and its (well established) transformation into a reward or cost based problem. We will then show how to transform a general RL problem back into a control problem [Rolf and Asada, 2014].

2.1 Reward Transformation

Adaptive motor control problems as shown in Fig. 1(a) follow a simple protocol: (1): The world provides a *goal* $\mathbf{x}^*$ to the agent that is situated in some *observation space* $\mathbb{X}\subseteq\mathbb{R}^n$. (2): The agent chooses an *action* $\mathbf{a}$ from some *action space* $\mathbb{A}\subseteq\mathbb{R}^m$. (3): The world provides a causal *outcome* $f(\mathbf{a})=\mathbf{x}$ of the agent's action, again situated in $\mathbb{X}$. The agent's task is to choose an action such that the outcome $\mathbf{x}$ matches the goal $\mathbf{x}^*$: $\mathbf{x}=f(\mathbf{a})=\mathbf{x}^*$. Many coordination problems provide *redundancy*: the action space is substantially higher dimensional than the observation space ($n \ll m$), such that multiple actions $\mathbf{a}_i \neq \mathbf{a}_j$ map to same outcome $f(\mathbf{a}_i)=f(\mathbf{a}_j)$. In such scenarios additional cost



(a) Adaptive motor control. (b) One-step reinforcement learning. (c) Latent goal analysis (LGA).

Figure 1: Latent goal analysis (LGA) identifies how to project actions and contexts into a common observation space. The observed rewards r are thereby explained by distance between action-outcome $\mathbf{x}$ (self-detection) and goal $\mathbf{x}^*$ (goal-detection), such that a reward problem is turned into a control problem.

functions $-e_a(\mathbf{a})$ are often used [Chung et al., 2007] to select an optimal action among those that fulfill $f(\mathbf{a}) = \mathbf{x}^*$. The ground truth function $f : \mathbb{A} \to \mathbb{X}$ is called *forward function*. This problem is often formalized by means of an overall cost-function of the distance of goal and outcome, and $e_a(\mathbf{a})$, which is easily transformed into reward semantics by inverting the sign. Apart from the self-detection $f(\mathbf{a})$, also the goals $\mathbf{x}^*$ are typically not simply given, but dynamically selected based on a larger system context. We can denote this selection on an abstract level with a function $h(\mathbf{c})$, which we refer to as *goal-detection*. Altogether this gives the reward transformation

$$r(\mathbf{c}, \mathbf{a}) = -||h(\mathbf{c}) - f(\mathbf{a})||^2 + e_c(\mathbf{c}) + e_a(\mathbf{a}) \ , \qquad (1)$$

where the virtual cost term $e_c(\mathbf{c})$ expresses that the quantity of reward can depend on the state, without effecting action selection. We will later utilize this term for theoretical considerations. The overall protocol now corresponds to a *one-step reinforcement problem* [Strehl, 2010, Langford and Zhang, 2008] as shown in Fig. 1(b): (1): The world provides a *context* $\mathbf{c}$ in some context space $\mathbb{C} \subseteq \mathbb{R}^p$. (2): The agent chooses an *action* $\mathbf{a}$ from the *action space* $\mathbb{A} \subseteq \mathbb{R}^m$. (3): The world provides a *reward* $r \in \mathbb{R}$ based on latent goals and action outcomes as in equation 1.

2.2 Latent Goal Transformation

So far we have established a formulation from existing goals to rewards. The idea for learning of goal representations is now to *invert* this process. Therefore we need to find functions $\hat{f}$, $\hat{h}$, $\hat{e}_c$ and $\hat{e}_a$ to resemble any possible reward function $r(\mathbf{c}, \mathbf{a})$:

$$r(\mathbf{c}, \mathbf{a}) = \hat{r}(\mathbf{c}, \mathbf{a}) = -||\hat{h}(\mathbf{c}) - \hat{f}(\mathbf{a})||^2 + \hat{e}_c(\mathbf{c}) + \hat{e}_a(\mathbf{a}) \ , \qquad (2)$$

or value function $Q(\mathbf{c}, \mathbf{a})$ expressing expected future rewards:

$$Q(\mathbf{c}, \mathbf{a}) = \hat{Q}(\mathbf{c}, \mathbf{a}) = -||\hat{h}(\mathbf{c}) - \hat{f}(\mathbf{a})||^2 + \hat{e}_c(\mathbf{c}) + \hat{e}_a(\mathbf{a})$$

This work does *not* tackle the temporal credit assignment problem to estimate Q itself. However, *if* a value system [Schultz et al., 1997, Daw and Doya, 2006] to estimate future rewards Q is already available, decomposing either a known estimate of $r(\mathbf{c}, \mathbf{a})$ or a known estimate of $Q(\mathbf{c}, \mathbf{a})$ is computationally equivalent since both are scalar functions of $\mathbf{c}$ / $\mathbf{a}$. The major challenge is to identify $\hat{f}(\mathbf{a}) = \hat{\mathbf{x}}$ and $\hat{h}(\mathbf{c}) = \hat{\mathbf{x}}^*$. Thereby goals and outcomes are considered *latent variables* of the reward function. These abstractions constitute the control problem in a low-dimensional observation space (see Fig. 1(c)). Cost terms depending on context *or* action only are considered as remainders, and in fact are easy to find given $\hat{f}$ and $\hat{h}$.

Ansatz Finding such functions can be formulated as finding appropriate coefficients of parametrized functions. First, we consider features $\psi_c(\mathbf{c}) : \mathbb{C} \to \mathbb{R}^{p'}$ and $\psi_a(\mathbf{a}) : \mathbb{A} \to \mathbb{R}^{m'}$ to describe the contexts and actions. Assuming an n-dimensional observation space $\mathbb{X}$ we can denote the function candidates with coefficients $\mathbf{M}$, $\mathbf{H}$, $\mathbf{R}_a$, and $\mathbf{R}_c$ as:

$$\begin{aligned}
\hat{\mathbf{x}}^* = \hat{h}(\mathbf{c}) &= \mathbf{H} \cdot \Psi_c(\mathbf{c}) \ , \quad \mathbf{H} \in \mathbb{R}^{n \times p'} \\
\hat{\mathbf{x}} = \hat{f}(\mathbf{a}) &= \mathbf{M} \cdot \Psi_a(\mathbf{a}) \ , \quad \mathbf{M} \in \mathbb{R}^{n \times m'} \\
\hat{e}_c(\mathbf{c}) &= \Psi_c(\mathbf{c})^T \cdot \mathbf{R}_c \cdot \Psi_c(\mathbf{c}) \ , \quad \mathbf{R}_c \in \mathbb{R}^{p' \times p'} \\
\hat{e}_a(\mathbf{a}) &= \Psi_a(\mathbf{a})^T \cdot \mathbf{R}_a \cdot \Psi_a(\mathbf{a}) \ , \quad \mathbf{R}_a \in \mathbb{R}^{m' \times m'} \ .
\end{aligned}$$

When we insert these definitions into Eqn. 2 we can write the reward transformation of a control problem in matrix notation

$$\hat{r}(\mathbf{c}, \mathbf{a}) = \begin{pmatrix} \psi_c(\mathbf{c}) \\ \psi_a(\mathbf{a}) \end{pmatrix}^T \begin{pmatrix} \mathbf{R}_c\text{-}\mathbf{H}^T\mathbf{H} & \mathbf{H}^T\mathbf{M} \\ \mathbf{M}^T\mathbf{H} & \mathbf{R}_a\text{-}\mathbf{M}^T\mathbf{M} \end{pmatrix} \begin{pmatrix} \psi_c(\mathbf{c}) \\ \psi_a(\mathbf{a}) \end{pmatrix}$$

as a quadratic form of context- and action-features.

Observation Space Reconstruction We can now write the *actual* reward function $r(\mathbf{c},\mathbf{a})$ similarly:

$$\begin{aligned} r(\mathbf{c},\mathbf{a}) &= \begin{pmatrix}\psi_c(\mathbf{c})\\ \psi_a(\mathbf{a})\end{pmatrix}^T \cdot \mathbf{K} \cdot \begin{pmatrix}\psi_c(\mathbf{c})\\ \psi_a(\mathbf{a})\end{pmatrix} \\ &= \begin{pmatrix}\psi_c(\mathbf{c})\\ \psi_a(\mathbf{a})\end{pmatrix}^T \begin{pmatrix}\mathbf{K}_{c,c} & \mathbf{K}_{c,a}\\ \mathbf{K}_{c,a}^T & \mathbf{K}_{a,a}\end{pmatrix} \begin{pmatrix}\psi_c(\mathbf{c})\\ \psi_a(\mathbf{a})\end{pmatrix} . \quad (3) \end{aligned}$$

This form with a symmetric coefficient matrix $\mathbf{K}$ is a *universal approximator*: it can arbitrarily well approximate at least all continuous functions if appropriate features ψ are chosen. For instance, if the features ψ_a and ψ_c are separate polynomial features of $\mathbf{a}$ and $\mathbf{c}$ up to polynomial degree d, then just the subterm $\psi_c(\mathbf{c})^T \cdot \mathbf{K}_{c,a} \cdot \psi_a(\mathbf{a})$ will contain all joint polynomial terms of $\mathbf{a}$ and $\mathbf{c}$ up to degree d. Hence, equation 3 can at least describe all functions that can be described by polynomials, i.e. all continuous functions.

We can now find coefficients $\mathbf{M}$, $\mathbf{H}$, $\mathbf{R}_a$, and $\mathbf{R}_c$ by matching equations 2.2 and 3. Starting from the need to match $\mathbf{K}_{c,a} = \mathbf{H}^T\mathbf{M}$, we can see that it is not only *always possible* to transform rewards into goals and outcomes, but it is even under-determined. There are infinitely many decompositions $\mathbf{H}^T\mathbf{M}$ for any matrix $\mathbf{K}_{c,a}$. For *any* choice of $\mathbf{H}$ and $\mathbf{M}$, a perfect match $r = \hat{r}$ can be generated by the residual terms

$$\mathbf{R}_c = \mathbf{K}_{c,c} + \mathbf{H}^T\mathbf{H} \quad \text{and} \quad \mathbf{R}_a = \mathbf{K}_{a,a} + \mathbf{M}^T\mathbf{M} .$$

For a concrete decomposition of $\mathbf{K}_{c,a}$ we can consider its singular value decomposition $\mathbf{U}\mathbf{S}\mathbf{V}^T$ with orthonormal matrices $\mathbf{U}, \mathbf{V}$ and a positive diagonal matrix $\mathbf{S}$. An exemplary decomposition could be to set $\mathbf{H} = \mathbf{S}^{\frac{1}{2}}\mathbf{U}^T$ and $\mathbf{M} = \mathbf{S}^{\frac{1}{2}}\mathbf{V}^T$. Still, the resulting observation space $\mathbb{X}$, in which $\hat{f}\colon \mathbb{A} \to \mathbb{X}$ and $\hat{h}\colon \mathbb{C} \to \mathbb{X}$ map actions and contexts, is very high-dimensional with $\min(p', m')$ dimensions, since $\mathbf{K}_{c,a} \in \mathbb{R}^{p' \times m'}$. However, a dimension reduction is now straightforward based on the SVD: we can select the diagonal matrix $\mathbf{S}' \in \mathbb{R}^{n \times n}$ with the n largest singular values of $\mathbf{K}_{c,a}$ and their respective singular vectors in $\mathbf{U}' \in \mathbb{R}^{p' \times n}$ and $\mathbf{V}' \in \mathbb{R}^{m' \times n}$ to approximate $\mathbf{K}_{c,a} \approx \mathbf{U}'\mathbf{S}'\mathbf{V}'^T = \mathbf{H}^T\mathbf{M}$. $\mathbf{H} \in \mathbb{R}^{n \times p'}$ and $\mathbf{M} \in \mathbb{R}^{n \times m'}$ can be chosen within the column space of $\mathbf{U}'$ and $\mathbf{V}'$ in order to project into the n-dimensional observation space. Hence, the latent observation space can be uniquely determined for any number of dimensions n. For sufficiently large n, LGA then approximates the reward function arbitrarily well. In order to make a concrete choice for $\mathbf{H}$ and $\mathbf{M}$ we choose a further criterion to minimize the the remainder terms $\hat{e}_c$ and $\hat{e}_a$. This can be operationalized by minimizing the respective matrix norms:

$$\begin{aligned} \mathbf{H}, \mathbf{M} &= \operatorname*{argmin}_{\mathbf{v},\mathbf{s}} \left(\|\mathbf{R}_c\|_2 + \|\mathbf{R}_a\|_2\right) \\ &\quad \text{such that } \mathbf{H}^T\mathbf{M} = \mathbf{K}_{c,a} . \end{aligned}$$

A method to perform this operation efficiently inside the already chosen n dimensional projection is described in [Rolf and Asada, 2014].

2.3 Algorithm and Interpretation

Altogether, LGA starts which a universal approximation of the reward or value function in the quadratic form shown in equation 3. The second step is the SVD of $\mathbf{K}_{c,a}$. Here we select the axis in column and row space that have the highest singular values. This corresponds to the axis of inside the action- and context space that are most significant to the reward/value function. Hence, this step identifies the low-dimensional observation space. The third step is to choose matrices $\mathbf{H}$ and $\mathbf{M}$ based on the criterion to minimize the remainder terms. This directly gives the functions $\hat{h}(\mathbf{c})$ and $\hat{f}(\mathbf{a})$ and allows to compute $\hat{e}_c(\mathbf{c})$ and $\hat{e}_a(\mathbf{a})$ if needed. $\hat{h}(\mathbf{c})$ and $\hat{f}(\mathbf{a})$ can then reduce the dimension of both states/contexts and actions.

3 News Article Recommendation

Our experiment investigates LGA's capability for dimension reduction in a one-step RL problem: a website comprising a certain set $A(t)$ of news articles at each time. One article can be featured at a prominent position on the website. The task is to select which article's teaser (action $\mathbf{a} \in A(t)$) should be featured. A *recommender system* is supposed to select these actions such that the probability that the website visitor interacts with it (e.g. clicks on the teaser) is maximized. In order to perform such selection specific to the visitor, there is information (context $\mathbf{c}$) available due to IP-address, or a login. With such information a reward function $r(\mathbf{c},\mathbf{a})$ can be estimated that resembles the click probability. Dimensionality reduction, however, is crucial in this domain: both context and action are typically very high-dimensional, but any recommender system must react extremely quickly to thousands or millions of visits. This can only be achieved if the dimension of $\mathbf{c}$ and $\mathbf{a}$ is reduced to allow for an efficient evaluation of $r(\mathbf{c},\mathbf{a})$.

3.1 Material and Method

For this experiment we use the "Yahoo! Front Page Today Module User Click Log Dataset, version 2.0", which comprises click recordings of *yahoo.com*'s front page from 15 consecutive days, from which we utilize the first day only. This recording contains $T = 1.6 \cdot 10^6$ events. Each event contains the actually displayed teaser, the set of currently available news, a set of visitor features, and the visitor's decision to click on the teaser ($r = 1$) or not ($r = 0$). 49 different teasers

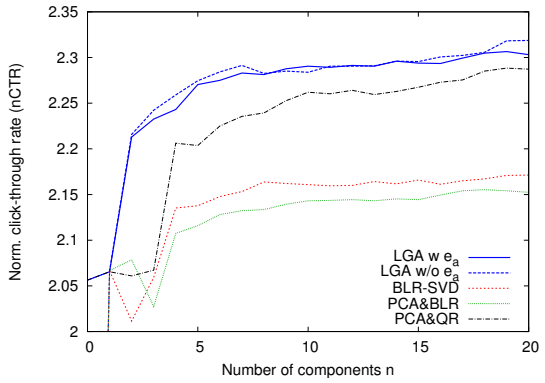


Figure 2: Results for the Yahoo! Webscope R6B data set. Estimated normalized click-through rates for LGA, BLR decomposition, and PCA depending on the number of selected components.

have been available and are represented as $m = 49$ dim. actions $\mathbf{a}$ encoded with a "1-of-m" scheme. The events contain 116 binary features about the visitor, which are *anonymized* in the data. We estimate $\mathbf{K}$ using batch-gradient descent on the empirical error $E[(\hat{r}(\mathbf{c},\mathbf{a},\mathbf{K})-r)^2]$. We applied 10000 epochs of training with a step width 0.01 starting from zero initial parameters. After that the parameters were fine-tuned by applying a whitening on the contexts and continuing batch regression for another 1000 epochs with step width 10^{-4}. As a baseline, we applied a bi-linear regression model $\hat{r}(\mathbf{c},\mathbf{a}) = \psi_c(\mathbf{c})^T \cdot \mathbf{B} \cdot \psi_a(\mathbf{a})$ that was trained with the same procedure. Such bi-linear regression (BLR) models have previously [Koren et al., 2009, Chu and Park, 2009] been used to reduce the dimension in recommender scenarios: the matrix $\mathbf{B}$ can be decomposed into $\mathbf{U}_B\mathbf{S}_B\mathbf{V}_B^T$ by singular value decomposition, after which only the n most significant dimensions are kept. The evaluation cost for both models is the same as both involve matrix-vector multiplications of the same size. As further baselines we used PCA to reduce the dimension of the context-space before applying either quadratic or bi-linear regression. In the PCA condition the dimension of actions cannot be reduced.

3.2 Results

For each method we can denote the policy to choose a news-teaser $\mathbf{a}$ based on the user information $\mathbf{c}$ as $\pi(\mathbf{c}) = \text{argmax}_{\mathbf{a}\in A(t)}\, \hat{r}(\mathbf{c},\mathbf{a})$, where $A(t)$ is the set of articles available at time t. A natural performance metric is the *click-through rate* $\text{CTR}_\pi = N_\pi^+/N$, where N is the total number of page visits and N_π^+ is the number of clicks generated by selecting teasers with π. Yet, this measure can only be measured when the policy is run *online* on the webpage. For an *offline evaluation* [Chu et al., 2009] we can estimate the performance by counting how often an actually clicked teaser would have been recommended by the policy:

$$\text{nCTR} = \frac{\text{CTR}_\pi}{\text{CTR}_\%} \approx \frac{|\{r_t = 1 \wedge \mathbf{a}_t = \pi(\mathbf{c}_t)\}|}{\sum_t (r_t \cdot |A(t)|^{-1})},$$

which is baselined against the performance $\text{CTR}_\%$ of a uniform random strategy. Fig. 2 shows that LGA achieves a substantially better performance than the bi-linear model decomposition (BLR-SVD). With rising n LGA quickly improves to nCTR>2.25 for $n\geq 5$ components and further improves to nCTR>2.3. The performance of LGA is largely unaffected by using the cost term $\hat{e}_a(\mathbf{a})$ or not. The bi-linear model requires $n \geq 8$ components to reach only nCTR > 2.15 with only minimal further improvement for more components. Both LGA and bi-linear decomposition outperform their counterparts with unsupervised PCA on the states before running bi-linear (BLR) or quadratic (QR) regression. Interestingly, PCA&QR substantially outperforms the standard BLR decomposition approach, which shows the high expressiveness of quadratic regression in general, but which comes with higher computational cost.

4 Discussion

We can conclude that LGA allows for an effective dimensionality reduction in the recommender setting, in which it outperforms the standard bi-linear model in terms of generated clicks. The margin is thereby numerically not very high in the range of 5-10%, but which is still highly significant to the domain since clicks are directly related to a website's monetary income. Other studies have reported much higher absolute values of CTR for other benchmarks, which suggests that the data set used here is rather hard. A possible reason is that there are no features for the actions, but only identities. A further interesting application (that would require non-anonymized data, though) would be to analyze the goal semantics in the observation space and see what kind of user features have been associated with which features of an article.

Acknowledgements

This study has been supported by the JSPS Grant-in-Aid for Specially promoted Research (No. 24000012). We also would like to thank the Yahoo! Webscope program for providing the R6B - Yahoo! Front Page Today Module User Click Log Dataset, version 2.0 for our experiments.

References

[Barto, 1994] Barto, A. G. (1994). Reinforcement learning in motor control. In *Handbook of Brain Theory and Neural Networks*, pages 809–813. Cambridge: MIT Press.

[Bitzer, 2011] Bitzer, S. (2011). Nonlinear dimensionality reduction for motion synthesis and control.

[Chu and Park, 2009] Chu, W. and Park, S.-T. (2009). Personalized recommendation on dynamic content using predictive bilinear models. In *Int. Conf. World wide web (WWW)*.

[Chu et al., 2009] Chu, W., Park, S.-T., Beaupre, T., Motgi, N., Phadke, A., Chakraborty, S., and Zachariah, J. (2009). A case study of behavior-driven conjoint analysis on yahoo!: front page today module. In *Int. Conf. Knowledge Discovery and Data Mining*.

[Chung et al., 2007] Chung, W., Fu, L.-C., and Hsu, S.-H. (2007). Chapter 6: Motion control. In Siciliano, B. and Khatib, O., editors, *Handbook of Robotics*, pages 133–160. Springer New York.

[Daw and Doya, 2006] Daw, N. D. and Doya, K. (2006). The computational neurobiology of learning and reward. *Current opinion in neurobiology*, 16(2):199–204.

[Kaelbling et al., 1996] Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). Reinforcement learning: A survey. *arXiv preprint cs/9605103*.

[Kober and Peters, 2012] Kober, J. and Peters, J. (2012). Reinforcement learning in robotics: A survey. In *Reinforcement Learning*, pages 579–610. Springer.

[Koren et al., 2009] Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37.

[Langford and Zhang, 2008] Langford, J. and Zhang, T. (2008). The epoch-greedy algorithm for contextual multi-armed bandits. In *NIPS*.

[Legenstein et al., 2010] Legenstein, R., Wilbert, N., and Wiskott, L. (2010). Reinforcement learning on slow features of high-dimensional input streams. *PLoS Computational Biology*, 6(8).

[Nguyen-Tuong and Peters, 2011] Nguyen-Tuong, D. and Peters, J. (2011). Model learning for robot control: a survey. *Cognitive Processing*, 12(4).

[Nouri and Littman, 2010] Nouri, A. and Littman, M. L. (2010). Dimension reduction and its application to model-based exploration in continuous spaces. *Machine Learning*, 81(1):85–98.

[Poupart and Boutilier, 2002] Poupart, P. and Boutilier, C. (2002). Value-directed compression of pomdps. In *NIPS*, pages 1547–1554.

[Rolf and Asada, 2014] Rolf, M. and Asada, M. (2014). Where do goals come from? A generic approach to autonomous goal-system development.

[Rolf and Steil, 2014] Rolf, M. and Steil, J. J. (2014). Explorative learning of inverse models: a theoretical perspective. *Neurocomputing*, 131:2–14.

[Rolf et al., 2010] Rolf, M., Steil, J. J., and Gienger, M. (2010). Goal babbling permits direct learning of inverse kinematics. *IEEE Trans. Autonomous Mental Development*, 2(3).

[Roy and Gordon, 2002] Roy, N. and Gordon, G. (2002). Exponential family pca for belief compression in pomdps. In *NIPS*, volume 2, pages 1043–1049.

[Schultz et al., 1997] Schultz, W., Dayan, P., and Montague, P. R. (1997). A neural substrate of prediction and reward. *Science*, 275(5306).

[Strehl, 2010] Strehl, A. L. (2010). Associative reinforcement learning. In *Encyclopedia of Machine Learning*, pages 49–51. Springer.

[Yahoo! Webscope, 2014] Yahoo! Webscope (2014). R6B - Yahoo! Front Page Today Module User Click Log Dataset, version 2.0. http://webscope.sandbox.yahoo.com.