Neighborhood Components Analysis for Reward-Based Dimensionality Reduction

Nathan Sprague NSPRAGUE@KZOO.EDU
Kalamazoo College, 1200 Academy Street, Kalamazoo, MI 49008 USA

There has been a great deal of research that attempts to explain the structure of biological receptive fields in terms of various methods for adapting basis vectors based on the statistical structure of visual input. These include principal components analysis (Hancock et al., 1992), independent components analysis (Bell & Sejnowski, 1997), non-negative matrix factorization (Lee & Seung, 1999), and predictive coding (Rao & Ballard, 1999), among others. Typically, such approaches are based purely on the structure of the visual input; there is no consideration of the role that visual information plays in the goal directed behavior of an organism. The motivation for the current work is to explore mechanisms of basis vector adaptation that are explicitly driven by the behavioral demands of a situated agent.

Our approach to addressing this issue is built on a body of recent work that attempts to automatically construct appropriate basis functions for reinforcement learning in continuous or high dimensional spaces (Keller et al., 2006; Mahadevan, 2005; Smart, 2004). Progress has been slow in developing RL algorithms that scale reliably from small discrete state spaces to high dimensional continuous spaces. One problem has been that scaling up traditional reinforcement learning algorithms, such as Q-Learning, requires replacing table based representations of the value function with function approximators. Unfortunately, the resulting algorithms lack convergence guarantees and often perform poorly in practice. One algorithm that addresses these issues is Least Squares Policy Iteration (LSPI) (Lagoudakis & Parr, 2003). LSPI couples a linear approximation architecture with approximate policy iteration. The result is an algorithm that is guaranteed not to diverge, and performs well in practice.

The effectiveness of LSPI is highly dependent upon selecting appropriate basis functions. In the work presented here, basis functions are applied in two steps: first high dimensional state vectors are projected to a lower dimensional space, then radial basis functions are applied to the lower dimensional data points. The focus here is on adapting the vectors used for dimensionality reduction. The radial basis functions are constructed by hand, and remain fixed throughout training.

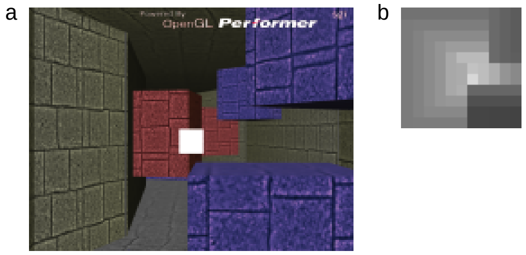

Figure 1. The Navigation Task. a) A view of the environment from behind the agent (white square). On each time step the agent chooses between three actions: forward-left forward-right or forward. He receives a positive reward on every time step that he successfully avoids colliding with obstacles or the walls. b) Depth information from the agent's point of view. The agents input takes the form of an 11x11 depth map of the area ahead. The two dark regions on the right represent looming obstacles.

The basis vectors used for dimensionality reduction are learned iteratively: First, LSPI is applied to learn a value function based on an arbitrarily chosen set of initial basis vectors. Sample points are then labeled with their one step Bellman update targets, and a modified version neighborhood components analysis (Keller et al., 2006; Goldberger et al., 2004) is used to find a new set of basis vectors that are better able to represent the target value function. This process is then iterated until convergence.

The algorithm has been tested on a simple three dimensional collision avoidance task in which an agent uses depth information to fly through a twisting corridor containing randomly positioned obstacles (Figure 1). The initial basis vectors are chosen to be the first two principal components of the training data (Figure 2a). After learning, the final basis vectors selectively extract information from the depth data that is relevant to the navigation task (Figures 2 and 3). Further, when the reward structure of the task is modified, in this case by vertically offsetting the agent's point of collision relative to his point of view, different basis vectors are discovered that are appropriate for the new version of the task, even though the statistics of the state information are unchanged (Figure 2c-d). Under all conditions, the policies learned using the adapted basis vectors are superior to the policies learned using the first two principle components (Figure 4) even though the principal components do a better job of capturing the variance in the state data.

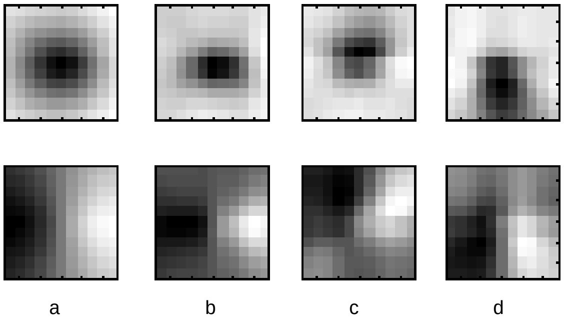

Figure 2. Basis Vectors. a) The first two principle components of the depth data. b) The basis vectors discovered by the iterated NCA algorithm for the navigation task described above. Note that, relative to the principal components, these vectors are mostly uniform along the top and bottom of the depth field. These areas correspond to regions of the environment that do not contain useful information for the collision avoidance task. c,d) The basis vectors discovered by the iterated NCA algorithm when the agent's point of collision is one meter above (c) or below (d) the center of his perceptual field. The structured part of the basis vector moves to the part of the depth field that could contain potential colliders.

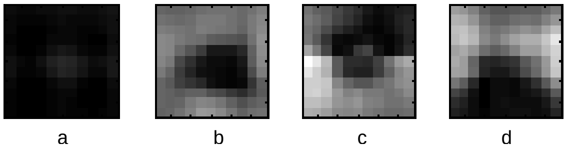

Figure 3. Per-Pixel Reconstruction Error. For this figure, the corresponding basis vectors from Figure 2 are orthonormalized and used to reconstruct the depth data from the training sets. Darker regions indicate lower mean squared reconstruction error. This illustrates that the basis vectors discovered by the iterated NCA algorithm preferentially represent regions of the depth images that are task relevant.

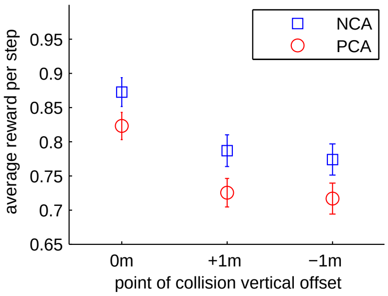

Figure 4. Average Reward. Average reward across 300 trials of 100 steps each. The horizontal axis represents the three different experimental conditions described in Figure 2. In each condition the average per-step reward is reported for policies learned using PCA basis vectors as well as policies learned using NCA basis vectors. One unit of reward is received for every step that does not result in collision. Therefore the y axis can be read as the percentage of steps that did not result in collision. Bars represent 95% confidence intervals.

References

Bell, A., & Sejnowski, T. (1997). The "independent components" of natural scenes are edge filters. *Vision Research*, *37*, 3327–38.

Goldberger, J., Roweis, S., Hinton, G., & Salakhutdinov, R. (2004). Neighbourhood component analysis. *Neural Information Processing Systems*.

Hancock, P. J., Baddeley, R. J., & Smith, L. S. (1992). The principal components of natural images. *Network*, *3*, 61–70.

Keller, P. W., Mannor, S., & Precup, D. (2006). Automatic basis function construction for approximate dynamic programming and reinforcement learning. *Proceedings of the 23rd international conference on Machine learning* (pp. 449 – 456). Pittsburgh, PA.

Lagoudakis, M. G., & Parr, R. (2003). Least-squares policy iteration. *Journal of Machine Learning Research*, *4*, 1107–1149.

Lee, D., & Seung, H. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*, 788–91.

Mahadevan, S. (2005). Samuel meets amarel: Automating value function approximation using global state space analysis. *Proceedings of the National Conference on Artificial Intelligence AAAI05*. Pittsburgh, PA.

Rao, R. P. N., & Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, *2*, 79 − 87.

Smart, W. (2004). Explicit manifold representations for value-function approximation in reinforcement learning. *Proceedings of the 8th International Symposium on AI and Mathematics*.