Feature Selection for Reinforcement Learning: Evaluating Implicit State-Reward Dependency via Conditional Mutual Information

Hirotaka Hachiya and Masashi Sugiyama

Tokyo Institute of Technology, Tokyo, 152-8552, Japan
hachiya@sg.cs.titech.ac.jp sugi@cs.titech.ac.jp

Abstract. Model-free reinforcement learning (RL) is a machine learning approach to decision making in unknown environments. However, real-world RL tasks often involve high-dimensional state spaces, and then standard RL methods do not perform well. In this paper, we propose a new feature selection framework for coping with high dimensionality. Our proposed framework adopts *conditional mutual information* between return and state-feature sequences as a feature selection criterion, allowing the evaluation of implicit state-reward dependency. The conditional mutual information is approximated by a least-squares method, which results in a computationally efficient feature selection procedure. The usefulness of the proposed method is demonstrated on grid-world navigation problems.

1 Introduction

Optimal decision making in unknown environment is a challenging task in the machine learning community. *Reinforcement learning* (RL) is a popular framework for this purpose, and has been actively studied. In RL, a *policy* (the decision rule of an agent) is determined so that *return* (the sum of discounted rewards the agent will receive) is maximized. So far, various RL approaches such as *policy iteration* [1, 2] and *policy search* [3–5] have been explored and demonstrated to be promising in small- to medium-sized problems.

However, when the dimensionality of the state space is high, existing RL approaches tends to perform poorly. Unfortunately, this critically limits the range of applicability of RL in practice since real-world RL tasks such as robot control often involve high-dimensional state spaces. To cope with high dimensionality of the state space, choosing a subset of relevant features from the high-dimensional state variables, i.e., *feature selection*, is highly useful.

For example, let us consider developing a security guard robot that can deal with a variety of tasks such as navigation, patrol, and intruder detection. For this purpose, the robot is equipped with various types of sensors such as position, orientation, distance, vision, sound, smell, and temperature sensors. However, when a security guard robot is engaged in a particular task such as navigation, all the sensors may not be needed. Since sensors necessary for solving a task are different depending on the tasks, it is not possible to choose the subset of sensors in advance. Thus, adaptive feature selection based on currently available data samples is indispensable in this scenario.

Various feature selection strategies have been explored so far, which can be categorized into the *wrapper* and *filter* approaches [6]. In the wrapper approach, features are selected depending on the subsequent learning process such as least-squares fitting of the *value function* [7–9]. A supervised dimensionality reduction method called *neighborhood component analysis* [10] was applied to feature selection in RL [7], while the decomposition of value function approximation error into reward prediction error and transition prediction error was utilized for feature selection in [9]. These wrapper methods would be useful for specific RL frameworks such as policy iteration, but they may not be directly employed in other frameworks such as policy search.

On the other hand, in the filter approach, features are selected independently of subsequent learning processes [11, 12]. More specifically, a subset of features is chosen in an *information-theoretic* way that the remaining subset of features is statistically independent of the outcome (typically, the rewards). Such an information-theoretic approach is versatile as preprocessing of high-dimensional data.

A supervised dimensionality reduction method called *kernel dimension reduction* (KDR) [13] was applied to feature selection in RL [11]. Based on the *Markov* property of RL problems, their method evaluates the conditional independence between the entire state features and a state subset which directly influences rewards at the next time-step. However, since RL deals with sequential decision making problems, there can exist an *implicit* dependency between state features and *rewards* through the process of sequential decision making. This is illustrated using a simple example in Figure 1. A feature $s^{(2)}$ influences $s^{(1)}$ at the next time-step which have a direct effect on rewards. Thus, features $s^{(1)}$ and $s^{(2)}$ can be selected by KDR. However, $s^{(3)}$ cannot be selected by KDR since there is no dependency between $s^{(1)}$ and $s^{(3)}$ in a single time-step, although it actually influences rewards in two time-steps through $s^{(2)}$ and $s^{(1)}$.

The implicit dependency in the sequential process can be detected in principle by recursively evaluating the dependency between states and rewards [12]. However, such a recursive approach is computationally demanding particularly when there exist *cascaded* dependency relations. For example, in Figure 1, two recursions are needed to find the relevant features $\{s^{(1)}, s^{(2)}, s^{(3)}\}$. First, $s^{(2)}$ is selected due to its dependency to $s^{(1)}$, and then $s^{(3)}$ is chosen because of its dependency to $s^{(2)}$. In addition, an assumption that the model of *factored* Markov decision processes is available as a dynamic Baysian network was imposed in [12], which may not be realistic.

In order to overcome the drawbacks of existing approaches, we introduce a new framework of filter-type feature selection for RL. More specifically, we propose to directly evaluate the independence between return and state-feature sequences using the *conditional mutual information* [14]. In order to efficiently approximate the conditional mutual information from samples, we utilize a least-

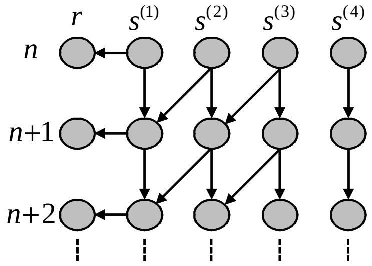


Fig. 1. An example of implicit dependency between state features and rewards. Each row represents the time-step $(n, n+1, n+2, \ldots)$, and columns represent reward(r) and state features ($s^{(1)}$, $s^{(2)}$, $s^{(3)}$, $s^{(4)}$). Arrows indicate the dependency between two variables; for example, an arrow from $s_n^{(2)}$ to $s_{n+1}^{(1)}$ exists if $s_{n+1}^{(1)}$ depends on $s_n^{(2)}$. In this example, a state feature $s^{(3)}$ does not have direct influence on the next reward r_n, but has indirect influence on r_{n+2} through $s_{n+1}^{(2)}$ and $s_{n+2}^{(1)}$.

squares (un-conditional) mutual information estimator which was proved to possess the optimal convergence rate [15].

The rest of this paper is organized as follows. In Section 2, we mathematically formulate the problem of RL. In Section 3, we describe our proposed feature selection procedure. Experimental results are reported in Section 4, demonstrating the effectiveness of the proposed method in grid-world navigation. Finally, we conclude in Section 5 by summarizing our contributions and describing future work.

2 Formulation of RL

In this section, we formulate the RL problem as a Markov decision process (MDP).

2.1 Markov Decision Process

Let us consider an MDP specified by $(\mathcal{S}, \mathcal{A}, p_{\mathrm{T}}, p_{\mathrm{I}}, R, \gamma)$, where $\mathcal{S}$ $(\in \mathbb{R}^v)$ is a set of v-dimensional states, $\mathcal{A}$ $(\in \mathbb{R})$ is a set of one-dimensional actions, $p_{\mathrm{T}}(\boldsymbol{s}'|\boldsymbol{s}, a)$ (≥ 0) is the transition probability-density from state $\boldsymbol{s}$ to next state $\boldsymbol{s}'$ when action a is taken, $p_{\mathrm{I}}(\boldsymbol{s})$ (≥ 0) is the probability density of initial states, $R(\boldsymbol{s}, a, \boldsymbol{s}')$ $(\in \mathbb{R})$ is an immediate reward for transition from $\boldsymbol{s}$ to $\boldsymbol{s}'$ by taking action a, and γ $(\in (0, 1])$ is the discount factor for future rewards. By following initial probability p_{I}, transition probability p_{T}, and policy π, an MDP generates a sequence of states, actions, and rewards as

$$\boldsymbol{s}_1, a_1, r_1, \boldsymbol{s}_2, a_2, r_2, \boldsymbol{s}_3, a_3, r_3, \ldots,$$

where the subscript indicates the time step. Let $p^\pi(\boldsymbol{s}|n)$ be the probability density of state $\boldsymbol{s}$ at time step n:

$$\begin{aligned}
p^\pi(\boldsymbol{s}|n=1) &= p_\mathrm{I}(\boldsymbol{s}),\\
p^\pi(\boldsymbol{s}|n=2) &= \iint p_\mathrm{I}(\boldsymbol{s}_1)\pi(a_1|\boldsymbol{s}_1)p_\mathrm{T}(\boldsymbol{s}|\boldsymbol{s}_1,a_1)\mathrm{d}\boldsymbol{s}_1\mathrm{d}a_1,\\
p^\pi(\boldsymbol{s}|n=3) &= \iiiint p_\mathrm{I}(\boldsymbol{s}_1)\pi(a_1|\boldsymbol{s}_1)p_\mathrm{T}(\boldsymbol{s}_2|\boldsymbol{s}_1,a_1)\\
&\qquad \times \pi(a_2|\boldsymbol{s}_2)p_\mathrm{T}(\boldsymbol{s}|\boldsymbol{s}_2,a_2)\mathrm{d}\boldsymbol{s}_1\mathrm{d}a_1\mathrm{d}\boldsymbol{s}_2\mathrm{d}a_2,\\
&\ \,\vdots
\end{aligned}$$

2.2 Optimal Policy

Let η_n $(\in \mathbb{R})$ be the *return* which is the sum of discounted rewards the agent will receive when starting from the n-th time step:

$$\eta_n \equiv \sum_{n'=n}^{\infty} \gamma^{n'-n} R(\boldsymbol{s}_{n'}, a_{n'}, \boldsymbol{s}_{n'+1}).$$

Let $p^\pi(\eta|\boldsymbol{s})$ be the probability density of return η when starting from a state $\boldsymbol{s}$ and then following a policy π. Let $V^\pi(\boldsymbol{s})$ be the *expected return*:

$$V^\pi(\boldsymbol{s}) \equiv \int \eta p^\pi(\eta|\boldsymbol{s})\mathrm{d}\eta.$$

The goal of RL is to learn the optimal policy π^* that maximizes the expected return $V^\pi(\boldsymbol{s})$:

$$\pi^*(\cdot|\boldsymbol{s}) \equiv \operatorname*{arg\,max}_{\pi(\cdot|\boldsymbol{s})} V^\pi(\boldsymbol{s}). \tag{1}$$

2.3 Data Samples

We suppose that a dataset consisting of M episodes of N steps is available. The agent initially starts from a randomly selected state $\boldsymbol{s}_1$ following the initial-state probability density $p_\mathrm{I}(\boldsymbol{s})$, and chooses an action based on a policy $\pi(a_n|\boldsymbol{s}_n)$. Then the agent makes a transition following $p_\mathrm{T}(\boldsymbol{s}_{n+1}|\boldsymbol{s}_n, a_n)$, and receives a reward r_n $(= R(\boldsymbol{s}_n, a_n, \boldsymbol{s}_{n+1}))$. This is repeated for N steps—thus the training data $\mathcal{D}^\pi$ is expressed as

$$\mathcal{D}^\pi \equiv \{d_n^\pi\}_{n=1}^N, \tag{2}$$

where each time-step data d_n^π consists of M sets of 3-tuple elements observed at each time step n as

$$d_n^\pi \equiv \{(\boldsymbol{s}_{m,n}^\pi, a_{m,n}^\pi, r_{m,n}^\pi)\}_{m=1}^M. \tag{3}$$

Let $\eta_{m,n}^\pi$ be the return in the m-th episode defined by

$$\eta_{m,n}^\pi \equiv \sum_{n'=n}^{N} \gamma^{n'-n} r_{m,n'}^\pi. \tag{4}$$

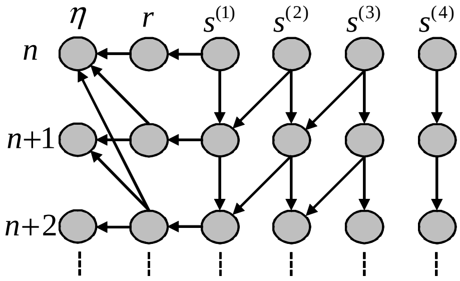


Fig. 2. An example of dependency between state features and returns. State features $s^{(1)}$, $s^{(2)}$ and $s^{(3)}$ influence returns at the same time-step, e.g., from $s_n^{(3)}$ to η_n.

3 Feature Selection via Conditional Mutual Information

In this section, we describe our proposed feature selection method.

Let η_n and $\boldsymbol{s}_n = (s_n^{(1)}, s_n^{(2)}, \ldots, s_n^{(v)})$ be the return and the state features at the n-th time step. For u $(\leq v)$ being the number of features we want to select, our goal is to find a 'subset' $\boldsymbol{z}_n = (z_n^{(1)}, z_n^{(2)}, \ldots, z_n^{(u)})^\top$ of the state features $\boldsymbol{s}_n$ such that

$$\eta_n \perp \boldsymbol{s}_n \mid \boldsymbol{z}_n, \quad \forall n = 1, 2, \ldots, N. \tag{5}$$

This means that, for all time steps, the return η_n is conditionally independent of the entire state features $\boldsymbol{s}_n$ given the subset $\boldsymbol{z}_n$.

The criterion (5) allows us to capture an indirect dependency from state features to rewards r since returns η contain all subsequent rewards. This is illustrated using a simple example in Figure 2. A state feature $s^{(3)}$ does not influence a reward r at the next time-step, but it does affect a return η at the same time-step since the return is the sum of discounted subsequent rewards, i.e., $\eta_n = r_n + \gamma r_{n+1} + \gamma^2 r_{n+2} + \cdots$.

3.1 Conditional Mutual Information

Mutual information (MI) is a popular measure of independence between random variables [14]. Here, we use a variant of MI based on the squared-loss [15] defined by

$$I^\pi(\eta_n; \mathrm{z}_n) \equiv \iint \left(\frac{p^\pi(\eta, \boldsymbol{z}|n)}{p^\pi(\eta|n) p^\pi(\boldsymbol{z}|n)} - 1 \right)^2 p^\pi(\eta|n) p^\pi(\boldsymbol{z}|n) \mathrm{d}\eta \mathrm{d}\boldsymbol{z}, \tag{6}$$

where $I^\pi(\eta_n; \mathrm{z}_n)$ denotes MI between return η_n and features $\boldsymbol{z}_n$ at the n-th time step when following a policy π. $p^\pi(\eta, \boldsymbol{z}|n)$ denotes the joint density of η and $\boldsymbol{z}$ at the n-th time step, and $p^\pi(\eta|n)$ and $p^\pi(\boldsymbol{z}|n)$ denote the marginal densities of return η and features $\boldsymbol{z}$ at the n-th time step, respectively. $I^\pi(\eta_n; \mathrm{z}_n)$ is non-negative and is equal to zero if and only if

$$p^\pi(\eta, \boldsymbol{z}|n) = p^\pi(\eta|n) p^\pi(\boldsymbol{z}|n),$$

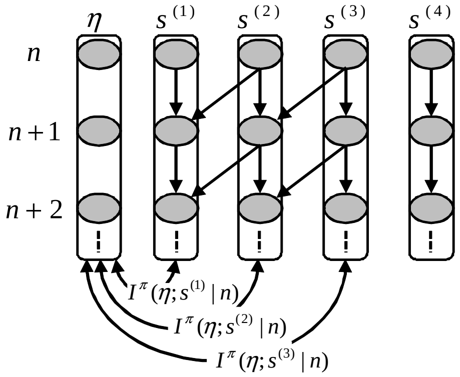

η
$s^{(1)}$
$s^{(2)}$
$s^{(3)}$
$s^{(4)}$
n
$n+1$
$n+2$
$I^{\pi}(\eta; s^{(1)} \mid n)$
$I^{\pi}(\eta; s^{(2)} \mid n)$
$I^{\pi}(\eta; s^{(3)} \mid n)$

Fig. 3. An example of dependency between returns and elements of states. State elements $s^{(1)}$, $s^{(2)}$, and $s^{(3)}$ directly influence the return η.

i.e., η and $\boldsymbol{z}$ are conditionally independent of each other given n.

We propose to use *conditional MI* $I^{\pi}(\eta; \mathrm{z}|\mathrm{n})$ as our feature selection criterion, which is defined as the average of MI $I^{\pi}(\eta_n; \mathrm{z}_n)$ over time steps $n = 1, 2, \ldots, N$ [14]:

$$I^{\pi}(\eta; \mathrm{z}|\mathrm{n}) = \frac{1}{N} \sum_{n=1}^{N} I^{\pi}(\eta_n; \mathrm{z}_n).$$

The conditional MI between returns and state features can be seen as a measure of dependency between returns and state-feature sequences as illustrated in Figure 3.

The rationale behind the use of conditional MI for feature selection relies on the following lemma (its proof is provided in Appendix.

Lemma 1.

$$\begin{aligned} I^{\pi}(\eta; \mathrm{s}|\mathrm{n}) - I^{\pi}(\eta; \mathrm{z}|\mathrm{n}) &= \frac{1}{N} \sum_{n=1}^{N} \iint \frac{p^{\pi}(\eta, \boldsymbol{z}|n)^2}{p^{\pi}(\eta|n) p^{\pi}(\boldsymbol{z}|n)^2} \\ &\quad \times \left(\frac{p^{\pi}(\eta, \boldsymbol{s}|\boldsymbol{z}, n)}{p^{\pi}(\boldsymbol{s}|\boldsymbol{z}, n) p^{\pi}(\eta|\boldsymbol{z}, n)} - 1 \right)^2 p^{\pi}(\boldsymbol{s}|n) \mathrm{d}\boldsymbol{s} \mathrm{d}\eta \\ &\geq 0. \end{aligned}$$

This lemma implies that $I^{\pi}(\eta; \mathrm{s}|\mathrm{n}) \geq I^{\pi}(\eta; \mathrm{z}|\mathrm{n})$ and the equality holds if and only if

$$p^{\pi}(\eta, \boldsymbol{s}|\boldsymbol{z}, n) = p^{\pi}(\eta|\boldsymbol{z}, n) p^{\pi}(\boldsymbol{s}|\boldsymbol{z}, n), \ \forall n = 1, 2, \ldots, N.$$

This is equivalent to Eq.(5), and thus Eq.(5) can be attained by maximizing $I^{\pi}(\eta; \mathrm{z}|\mathrm{n})$ with respect to $\boldsymbol{z}$.

3.2 Estimation of Conditional Mutual Information

Since $I^{\pi}(\eta; \mathrm{z}|\mathrm{n})$ is not accessible, it needs to be estimated from data samples. Here, we employ a recently-proposed MI estimator called *least-squares MI* (LSMI) [15] for approximating the conditional MI $I^{\pi}(\eta; \mathrm{z}|\mathrm{n})$. A MATLAB® implementation of LSMI is available from

```
http://sugiyama-www.cs.titech.ac.jp/~sugi/software/LSMI/
```

The basic idea of LSMI is to estimate the ratio of probability densities $w_n(\eta, \boldsymbol{z}) \equiv \frac{p^{\pi}(\eta,\boldsymbol{z}|n)}{p^{\pi}(\eta|n)p^{\pi}(\boldsymbol{z}|n)}$ contained in MI without going through density estimation of $p^{\pi}(\eta, \boldsymbol{z}|n)$, $p^{\pi}(\eta|n)$, and $p^{\pi}(\boldsymbol{z}|n)$. Since density estimation is known to be a hard task [16], avoiding density estimation and directly estimating their ratio would be preferable [17, 18].

The density ratio function $w_n(\eta, \boldsymbol{z})$ is approximated by the following linear model:

$$\widehat{w}_n(\eta, \boldsymbol{z}) \equiv \boldsymbol{\alpha}_n^{\top} \boldsymbol{\psi}_n(\eta, \boldsymbol{z}),$$

where $\boldsymbol{\alpha}_n = (\alpha_{n,1}, \alpha_{n,2}, \dots, \alpha_{n,B})^{\top}$ are parameters to be learned, B is the number of parameters, and

$$\boldsymbol{\psi}_n(\eta, \boldsymbol{z}) = (\psi_{n,1}(\eta, \boldsymbol{z}), \psi_{n,2}(\eta, \boldsymbol{z}), \dots, \psi_{n,B}(\eta, \boldsymbol{z}))^{\top}$$

are basis functions such that

$$\psi_{n,b}(\eta, \boldsymbol{z}) \geq 0, \ \forall b, \ \forall (\eta, \boldsymbol{z}).$$

The parameter $\boldsymbol{\alpha}_n$ is determined so that the following squared error J_0 is minimized:

$$\begin{aligned} J_0(\boldsymbol{\alpha}_n) &\equiv \frac{1}{2} \iint \left(\widehat{w}_n(\eta, \boldsymbol{z}) - w_n(\eta, \boldsymbol{z})\right)^2 p^{\pi}(\eta|n) p^{\pi}(\boldsymbol{z}|n) \mathrm{d}\eta \mathrm{d}\boldsymbol{z} \\ &= \frac{1}{2} \iint \widehat{w}_n^2(\eta, \boldsymbol{z}) p^{\pi}(\eta|n) p^{\pi}(\boldsymbol{z}|n) \mathrm{d}\eta \mathrm{d}\boldsymbol{z} \\ &\quad - \iint \widehat{w}_n(\eta, \boldsymbol{z}) p^{\pi}(\eta, \boldsymbol{z}|n) \mathrm{d}\eta \mathrm{d}\boldsymbol{z} + C, \end{aligned}$$

where

$$C \equiv \frac{1}{2} \iint w_n^2(\eta, \boldsymbol{z}) p^{\pi}(\eta, \boldsymbol{z}|n) \mathrm{d}\eta \mathrm{d}\boldsymbol{z}$$

is a constant and thus can be safely ignored. Let us denote the first two terms by J:

$$\begin{aligned} J(\boldsymbol{\alpha}_n) &\equiv J_0(\boldsymbol{\alpha}_n) - C \\ &= \frac{1}{2} \boldsymbol{\alpha}_n^{\top} \boldsymbol{H}_n \boldsymbol{\alpha}_n - \boldsymbol{h}_n^{\top} \boldsymbol{\alpha}_n, \end{aligned} \tag{7}$$

where

$$\boldsymbol{H}_n \equiv \iint \boldsymbol{\psi}_n(\eta, \boldsymbol{z})\boldsymbol{\psi}_n(\eta, \boldsymbol{z})^\top p^\pi(\eta|n)p^\pi(\boldsymbol{z}|n)\mathrm{d}\eta\mathrm{d}\boldsymbol{z},$$
$$\boldsymbol{h}_n \equiv \iint \boldsymbol{\psi}_n(\eta, \boldsymbol{z})p^\pi(\eta, \boldsymbol{z}|n)\mathrm{d}\eta\mathrm{d}\boldsymbol{z}.$$

The expectations in $\boldsymbol{H}_n$ and $\boldsymbol{h}_n$ are approximated by the empirical averages using a one-step data sample d_n^π (see Eq.(3)).

$$\widehat{\boldsymbol{H}}_n \equiv \frac{1}{M^2}\sum_{m,m'=1}^{M} \boldsymbol{\psi}_n(\eta_{m,n}, \boldsymbol{z}_{m',n})\boldsymbol{\psi}_n(\eta_{m,n}, \boldsymbol{z}_{m',n})^\top,$$
$$\widehat{\boldsymbol{h}}_n \equiv \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{\psi}_n(\eta_{m,n}, \boldsymbol{z}_{m,n}).$$

Then the following optimization problem is obtained:

$$\widehat{\boldsymbol{\alpha}}_n \equiv \mathop{\arg\min}_{\boldsymbol{\alpha}_n \in \mathbb{R}^B} \left[\frac{1}{2}\boldsymbol{\alpha}_n^\top \widehat{\boldsymbol{H}}_n \boldsymbol{\alpha}_n - \widehat{\boldsymbol{h}}_n^\top \boldsymbol{\alpha}_n + \frac{\lambda}{2}\boldsymbol{\alpha}_n^\top \boldsymbol{\alpha}_n\right], \tag{8}$$

where a regularization term $\lambda\boldsymbol{\alpha}_n^\top\boldsymbol{\alpha}_n/2$ is included. Differentiating the above objective function with respect to $\boldsymbol{\alpha}_n$ and equating it to zero, the solution can be obtained analytically as

$$\widehat{\boldsymbol{\alpha}}_n = (\widehat{\boldsymbol{H}}_n + \lambda\boldsymbol{I}_B)^{-1}\widehat{\boldsymbol{h}}_n,$$

where I_B denotes the B-dimensional identity matrix.

Using a density ratio estimator $\widehat{\boldsymbol{\alpha}}_n^\top\boldsymbol{\psi}_n(\eta, \boldsymbol{z})$, we can construct a conditional MI estimator between return and state-feature sequences as

$$\widehat{I}^\pi(\eta; \mathrm{z}|\mathrm{n}) \equiv \frac{1}{N}\sum_{n=1}^{N} \widehat{I}^\pi(\eta_n; \mathrm{z}_n), \tag{9}$$

where $\widehat{I}^\pi(\eta_n; \mathrm{z}_n)$ is an MI estimator between returns and state features at the n-th time step given as

$$\widehat{I}^\pi(\eta_n; \mathrm{z}_n) \equiv \frac{1}{M^2}\sum_{m,m'=1}^{M} \left(\widehat{\boldsymbol{\alpha}}_n^\top\boldsymbol{\psi}_n(\eta_{m,n}, \boldsymbol{z}_{m',n}) - 1\right)^2. \tag{10}$$

3.3 Feature Selection Algorithm

Finally, we describe how features are selected based on the conditional MI estimator $\widehat{I}^\pi(\eta; \mathrm{z}|\mathrm{n})$.

Forward selection and *backward elimination* would be two major strategies of feature selection [6, 19]. Here we employ forward selection since it was computationally more efficient and performed well in our preliminary experiments.

```
Algorithm 1: ForwardSelection($u, \mathcal{D}^\pi$)
  //$u$   : Number of features we want to choose
  //$\mathcal{D}^\pi$ : Data samples collected following $\pi$
  //$v$   : Number of all features
  //$\mathcal{I}$   : Remaining feature indices
  //$\mathcal{J}$   : Chosen feature indices
  $\mathcal{I} \leftarrow \{1, 2, \ldots, v\}$
  $\mathcal{J} \leftarrow \{\}$
  for $u' = 1, 2, \ldots, u$
      // Find the feature that maximizes the conditional mutual information
      $k \leftarrow \arg\max_{i \in \mathcal{I}} \sum_{n=1}^{N} \mathrm{LSMI}(d_n^\pi, \{s_n^{(j)}\}_{j \in \mathcal{J}} \cup s_n^{(i)})$
      $\mathcal{I} \leftarrow \mathcal{I} \backslash k$        // Remove $k$ from $\mathcal{I}$
      $\mathcal{J} \leftarrow \mathcal{J} \cup k$      // Add $k$ to $\mathcal{J}$
  return $(\mathcal{J})$
```

Fig. 4. A pseudo code of the proposed feature selection algorithm with forward selection. By the LSMI function, MI between return and state features is computed using the n-th time step data d_n^π.

Let $\mathcal{J}$ be the set of chosen feature indices. The forward selection algorithm starts from the empty feature-index set $\mathcal{J} = \{\}$. The index of the most relevant feature, together with features whose indices are included in $\mathcal{J}$, is sequentially added to $\mathcal{J}$ at each iteration. The relevance of each state feature $s^{(i)}$ is evaluated using the conditional MI estimator $\widehat{I}^\pi(\eta; \mathrm{z}|\mathrm{n})$ described in Section 3.2. This forward selection process is repeated u times, where u is the number of features we want to choose.

A pseudo code of the proposed feature selection algorithm with forward selection is described in Figure 4.

4 Numerical Experiments

In this section, we evaluate the performance of our proposed feature selection method on a grid-world navigation problem illustrated in Figure 5. The two-dimensional maze consists of walls (black cells) and target states (light-gray cells). The goal of the task is to navigate an agent to the target location by avoiding the walls.

4.1 Setup

The state space $\mathcal{S}$ consists of 14-dimensional discrete features $\boldsymbol{s} = (s^{(1)}, s^{(2)}, \ldots, s^{(14)})^\top$, where $s^{(1)}, s^{(2)} \in \{1, 2, \ldots, 10\}$ are the horizontal and vertical positions of the agent, respectively. $s^{(3)} \in \{0, 1, 2, \ldots, 20\}$ is the remaining battery level; it is initially set to 20 (fully charged) and is decreased by 1 at

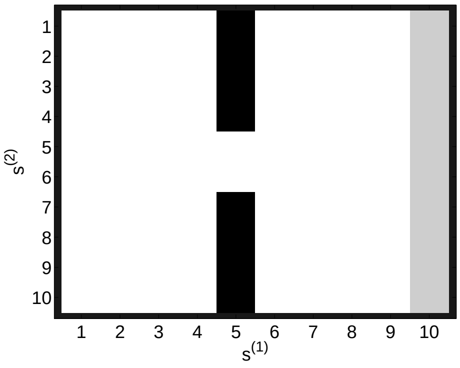


Fig. 5. A grid-world navigation problem. An agent is placed randomly in the grid-world, and can move around in the white area based on the four actions: moving up, down, left, or right at every time step. Black boxes in the middle represent walls through which the agent cannot go, and target location to which we want to guide the agent is the light-gray area at $s^{(1)} = 10$.

every agent's movement. The rest of features $s^{(4)}, s^{(5)}, s^{(6)}, \ldots, s^{(14)}$ corresponds to noise, each of which independently follows the Gaussian distribution with different mean:

$$\frac{1}{\sigma_{\text{noise}}\sqrt{2\pi}} \exp\left(-\frac{(s^{(i)} - \nu_i)^2}{2\sigma_{\text{noise}}^2}\right), \quad \forall i = 4, 5, 6, \ldots, 14.$$

We set $\nu_i = i - 3$ and $\sigma_{\text{noise}} = 1$, and round the value of $s^{(i)}$ down to the nearest integer for discretization. These additional dimensions of the state space may be regarded as information brought by irrelevant sensors such as sound, smell, and temperature sensors.

The action space $\mathcal{A}$ consists of four discrete actions, each of which corresponds to the direction of the agent's move: up, down, left, and right. For instance, if the agent chooses the 'right'-action, $s^{(1)}$ is incremented unless there is an wall and the battery level $(s^{(3)})$ is zero.

The reward $+2$ is given when the agent visits the target location; otherwise the reward is zero:

$$R(\boldsymbol{s}, a, \boldsymbol{s}') = \begin{cases} 2 & \text{if } s'^{(1)} = 10, \\ 0 & \text{otherwise.} \end{cases}$$

The discount factor is set to $\gamma = 0.95$.

Data samples $\mathcal{D}^{\pi}$ consisting M episodes with $N = 20$ steps are collected. The initial position of the agent is set to

$$(\boldsymbol{s}_1^{(1)}, \boldsymbol{s}_1^{(2)}) = (1, \beta),$$

where β is randomly chosen from $\{1, 2, \ldots, 10\}$. Then, the agent follows a stochastic policy $\pi(a|\boldsymbol{s})$ defined by

$$\pi(a|\boldsymbol{s}) = \begin{cases} 0.7 & \text{if } a = a^*, \\ 0.1 & \text{otherwise.} \end{cases}$$

where a^* is 'down' when $s^{(1)} = 4$ and $1 \leq s^{(2)} \leq 4$, a^* is 'up' when $s^{(1)} = 4$ and $7 \leq s^{(2)} \leq 10$, and a^* is 'right' in other states. We compute the conditional MI estimator $\widehat{I}^\pi(\eta; \mathrm{z}|\mathrm{n})$ from the dataset $\mathcal{D}^\pi$ (see Section 3.2). Gaussian kernels are used as basis functions:

$$\psi_{n,b}(\eta, \boldsymbol{z}) \equiv \exp\left(-\frac{\|(\eta, \boldsymbol{z}^\top)^\top - \boldsymbol{\mu}_{n,b}\|^2}{2\sigma_n^2}\right),$$

where $\boldsymbol{\mu}_{n,b}$ and σ_n are the mean and standard deviation of the Gaussian kernel, respectively. We set $B = M$, i.e., the number B of basis functions is equal to the number M of episodes. The mean $\boldsymbol{\mu}_{n,b}$ is selected from the dataset d_n^π as $\boldsymbol{\mu}_{n,b} = (\eta_{n,b}^\pi, \boldsymbol{z}_{n,b}^{\pi\top})^\top$. The standard deviation σ_n as well as the regularization parameter λ (see Eq.(8)) is determined by cross-validation with respect to J (see Eq.(7)) [15].

We compare the performance of our proposed method with the KDR-based method [11]. The feature selection criterion used by the KDR-based method is the conditional independence between states $\boldsymbol{s}$ and its subset $\boldsymbol{s}^r$ which directly influences rewards:

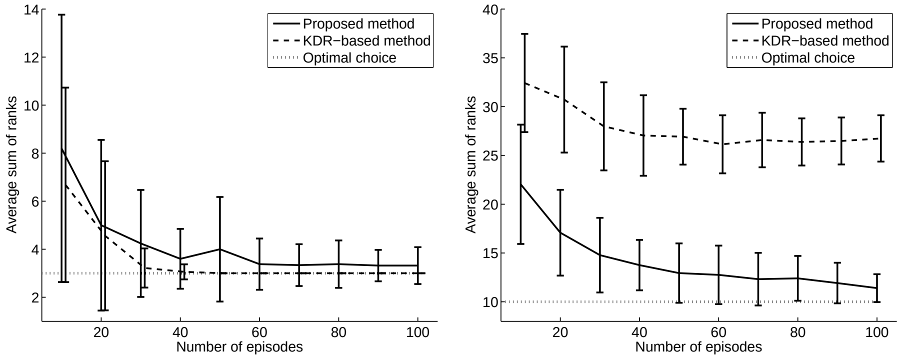
$$\boldsymbol{s}_{n+1}^r \perp \boldsymbol{s}_n \mid \boldsymbol{z}_n, \quad \forall n = 1, 2, \ldots, N,$$

where $\boldsymbol{s}^r = \{s^{(1)}\}$ in the current navigation problem. This criterion is evaluated using the *conditional cross-covariance* operator in a Gaussian reproducing kernel Hilbert space.

Similarly to our proposed method, we implement the KDR-based method based on the forward selection strategy: starting from the empty set $\mathcal{J} = \{\}$, the index of the state feature $s^{(i)}$, which, together with features whose indices are included in $\mathcal{J}$, attains the above conditional independence the most is added to $\mathcal{J}$ at every iteration. Following the suggestion in [13], we fix the width of the Gaussian kernel to the median of the distance between all the data samples, and fix the regularization parameter to 0.1.

To illustrate how the feature selection methods work in our grid-world navigation task, we run the forward selection algorithms for 14 iterations to rank all the state features. Let us consider the following two cases: the "without-gravel" and "with-gravel" scenarios. In the "without-gravel" scenario, state features $s^{(1)}$ and $s^{(2)}$ should have higher ranks because the horizontal position of the agent $s^{(1)}$ determines the reward directly and its vertical position $s^{(2)}$ is necessary to avoid walls in the middle of the maze.

On the other hand, in the "with-gravel" scenario, gravel exists in some grids and the state feature $s^{(4)}$ detects the existence of gravel; when $s^{(4)} = 2$, there



(a) "Without-gravel" scenario (b) "With-gravel" scenario

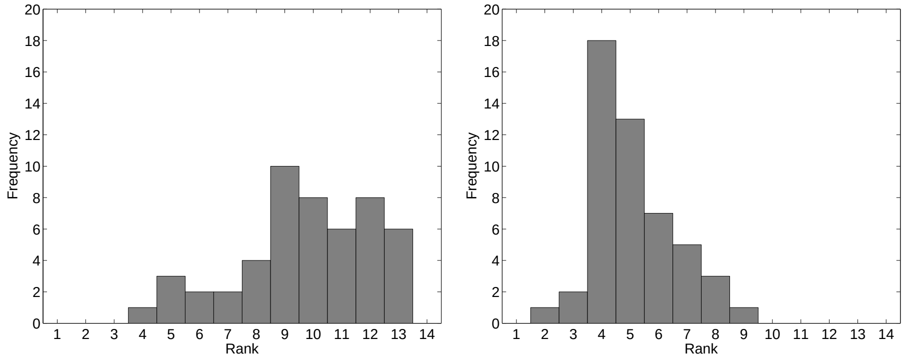
Fig. 6. Feature selection performance in the grid-world navigation task. The graphs depict the sum of ranks of relevant features averaged over 50 trials as a function of the number M of episodes. The optimal values are 3 (=1 + 2) and 10 (= 1 + 2 + 3 + 4) in the "without-gravel" and "with-gravel" scenarios, respectively.

exists gravel in the right grid of the agent. The agent can avoid the gravel area by moving to the left when $s^{(4)} = 2$; otherwise, it gets into the gravel area which continues for 4 time steps (this is indicated by $s^{(4)} = 3$). When the agent is in the gravel area, the battery level $s^{(3)}$ is decreased by 3 at each step. Then, the agent can not be able to reach the target place due to lack of battery (recall that the battery level is decreased by 1 at each step outside the gravel area). This indicates that the gravel-feature $s^{(4)}$ indirectly influences rewards after several time steps through the battery level $s^{(3)}$ and the horizontal position $s^{(1)}$. Therefore, in this case, in addition to $s^{(1)}$ and $s^{(2)}$, $s^{(3)}$ and $s^{(4)}$ should also have higher ranks to avoid the gravel area.

4.2 Results

Figure 6(a) depicts the sum of ranks of the features $s^{(1)}$ and $s^{(2)}$ averaged over 50 trials as a function of the number M of episodes for the "without-gravel" scenario. Since the features $s^{(1)}$ and $s^{(2)}$ should be ranked first and second, the optimal value is 3 (= 1+2). The graph overall shows that the performance of both the proposed and KDR-based methods improves as the number M of episodes increases. The KDR-based method converges to the optimal value (= 3) at 40 episodes, while a small error remains in the proposed method. This difference is caused by the fact that the battery level $s^{(3)}$ is occasionally ranked higher than the position $s^{(1)}$ and $s^{(2)}$ in the proposed method. Since the battery level is also somewhat relevant to returns, the result of the proposed method would be reasonable.

Figure 6(b) depicts the sum of ranks of the features $s^{(1)}$, $s^{(2)}$, $s^{(3)}$, and $s^{(4)}$ averaged over 50 trials as a function of the number M of episodes for the "with-gravel" scenario. Unlike the case of the "without-gravel" scenario, the per-



(a) KDR-based method

(b) Proposed method

Fig. 7. Histograms of the rank of the feature $s^{(4)}$ in the "with-gravel" scenario. The number M of episodes is fixed to 100 and the number of trials is 50.

formance improvement of the KDR-based method is very slow and is saturated after 60 episodes. This happened because evaluating the one-step dependency from the gravel feature $s^{(4)}$ to the horizontal position $s^{(1)}$ cannot find the relevance of $s^{(4)}$ to subsequent rewards.

Figure 7(a) depicts the histogram of the rank of the gravel-feature $s^{(4)}$ in the "with-gravel" scenario when the number M of episodes is fixed to 100. The graph shows that the KDR-based method ranks $s^{(4)}$ in lower positions, particularly in the range between 8 and 13. This implies that the feature $s^{(4)}$ is less frequently selected by the KDR-based method and then the gravel cannot be avoided properly.

On the other hand, the performance of the proposed method improves as the number M of episodes increases, and approaches the optimal value ($= 10$) (see Figure 6(b)). Figure 7(b)) shows that the proposed method ranks $s^{(4)}$ in higher positions, particularly around 4. This was achieved because the proposed method evaluates the dependency between returns η and $s^{(4)}$.

Overall, the proposed method was shown to be a promising feature selection method in RL.

5 Conclusions

In real-world reinforcement learning problems, selecting a subset of relevant attributes from the high-dimensional state variable is considerably important since standard reinforcement learning methods do not perform well with high-dimensional state spaces. An existing feature selection approach relies on the conditional independence between state and its subset which directly influences rewards. However, this is not appropriate when there is an indirect dependency between states and rewards, which is often the case in practical RL scenarios.

To overcome this limitation, we proposed a new framework of feature selection by considering the dependency between return and state-feature sequences. Our framework adopts conditional mutual information as the dependency measure, and it is approximated using least-squares estimation. The effectiveness of the proposed method was shown through experiments.

Acknowledgment

The authors were supported by the FIRST program.

Appendix: Proof of Lemma 1

Here, we give a proof of Lemma 1.

From the definition of conditional mutual information, we have

$$
\begin{aligned}
&I^\pi(\eta;\mathrm{s}|\mathrm{n}) - I^\pi(\eta;\mathrm{z}|\mathrm{n}) \\
&= \frac{1}{N}\sum_{n=1}^{N}\int p^\pi(\eta|n)\left\{\int\left(\frac{p^\pi(\eta,\boldsymbol{s}|n)}{p^\pi(\eta|n)p^\pi(\boldsymbol{s}|n)}-1\right)^2 p^\pi(\boldsymbol{s}|n)\mathrm{d}\boldsymbol{s}\right. \\
&\qquad\qquad\left.-\int\left(\frac{p^\pi(\eta,\boldsymbol{z}|n)}{p^\pi(\eta|n)p^\pi(\boldsymbol{z}|n)}-1\right)^2 p^\pi(\boldsymbol{z}|n)\mathrm{d}\boldsymbol{z}\right\}\mathrm{d}\eta \\
&= \frac{1}{N}\sum_{n=1}^{N}\int p^\pi(\eta|n)\left\{\int\left(\frac{p^\pi(\eta,\boldsymbol{s}|n)}{p^\pi(\eta|n)p^\pi(\boldsymbol{s}|n)}\right)^2 p^\pi(\boldsymbol{s}|n)\mathrm{d}\boldsymbol{s} - 2\int\frac{p^\pi(\eta,\boldsymbol{s}|n)}{p^\pi(\eta|n)}\mathrm{d}\boldsymbol{s}\right. \\
&\qquad\qquad\left.-\int\left(\frac{p^\pi(\eta,\boldsymbol{z}|n)}{p^\pi(\eta|n)p^\pi(\boldsymbol{z}|n)}\right)^2 p^\pi(\boldsymbol{z}|n)\mathrm{d}\boldsymbol{z} + 2\int\frac{p^\pi(\eta,\boldsymbol{z}|n)}{p^\pi(\eta|n)}\mathrm{d}\boldsymbol{z}\right\}\mathrm{d}\eta \\
&= \frac{1}{N}\sum_{n=1}^{N}\int p^\pi(\eta|n)\left\{\int\left(\frac{p^\pi(\eta,\boldsymbol{s}|n)}{p^\pi(\eta|n)p^\pi(\boldsymbol{s}|n)}\right)^2 p^\pi(\boldsymbol{s}|n)\mathrm{d}\boldsymbol{s}\right. \\
&\qquad\qquad\left.-\int\left(\frac{p^\pi(\eta,\boldsymbol{z}|n)}{p^\pi(\eta|n)p^\pi(\boldsymbol{z}|n)}\right)^2 p^\pi(\boldsymbol{z}|n)\mathrm{d}\boldsymbol{z}\right\}\mathrm{d}\eta \\
&= \frac{1}{N}\sum_{n=1}^{N}\int p^\pi(\eta|n)\int\left(\frac{p^\pi(\eta,\boldsymbol{s}|n)}{p^\pi(\eta|n)p^\pi(\boldsymbol{s}|n)}-\frac{p^\pi(\eta,\boldsymbol{z}|n)}{p^\pi(\eta|n)p^\pi(\boldsymbol{z}|n)}\right)^2 p^\pi(\boldsymbol{s}|n)\mathrm{d}\boldsymbol{s}\mathrm{d}\eta.
\end{aligned}
$$

To obtain the second equality above, we used

$$
\int\frac{p^\pi(\eta,\boldsymbol{s}|n)}{p^\pi(\eta|n)}\mathrm{d}\boldsymbol{s} = \iint\frac{p^\pi(\eta,\boldsymbol{z},\bar{\boldsymbol{z}}|n)}{p^\pi(\eta|n)}\mathrm{d}\boldsymbol{z}\mathrm{d}\bar{\boldsymbol{z}} = \int\frac{p^\pi(\eta,\boldsymbol{z}|n)}{p^\pi(\eta|n)}\mathrm{d}\boldsymbol{z},
$$

where $\bar{\boldsymbol{z}}$ is the complement of $\boldsymbol{z}$. To obtain the third equality, we used

$$
\begin{aligned}
&\int\left(\frac{p^\pi(\eta,\boldsymbol{z}|n)}{p^\pi(\eta|n)p^\pi(\boldsymbol{z}|n)}\right)^2 p^\pi(\boldsymbol{z}|n)\mathrm{d}\boldsymbol{z} \\
&\qquad= \int\frac{p^\pi(\eta,\boldsymbol{s}|n)p^\pi(\eta,\boldsymbol{z}|n)}{p^\pi(\eta|n)^2 p^\pi(\boldsymbol{s}|n)p^\pi(\boldsymbol{z}|n)}p^\pi(\boldsymbol{s}|n)\mathrm{d}\boldsymbol{s}.
\end{aligned}
$$

Since

$$
\begin{aligned}
p^{\pi}(\boldsymbol{s}|\boldsymbol{z},n)p^{\pi}(\boldsymbol{z}|n) &= p^{\pi}(\boldsymbol{s},\boldsymbol{z}|n) = p^{\pi}(\boldsymbol{s}|n),\\
\frac{p^{\pi}(\eta,\boldsymbol{z}|n)}{p^{\pi}(\eta|\boldsymbol{z},n)p^{\pi}(\boldsymbol{z}|n)} &= 1,\\
\frac{p^{\pi}(\eta,\boldsymbol{s}|n)}{p^{\pi}(\eta|n)p^{\pi}(\boldsymbol{s}|n)} &= \frac{p^{\pi}(\eta,\boldsymbol{s}|n)p^{\pi}(\eta,\boldsymbol{z}|n)}{p^{\pi}(\boldsymbol{s}|\boldsymbol{z},n)p^{\pi}(\eta|\boldsymbol{z},n)p^{\pi}(\eta|n)p^{\pi}(\boldsymbol{z}|n)},
\end{aligned}
$$

we have

$$
\begin{aligned}
&I^{\pi}(\eta;\mathrm{s}|\mathrm{n}) - I^{\pi}(\eta;\mathrm{z}|\mathrm{n})\\
&= \frac{1}{N}\sum_{n=1}^{N}\int p^{\pi}(\eta|n)\int \frac{p^{\pi}(\eta,\boldsymbol{z}|n)^2}{p^{\pi}(\eta|n)^2 p^{\pi}(\boldsymbol{z}|n)^2}\\
&\qquad\qquad \times\left(\frac{p^{\pi}(\eta,\boldsymbol{s}|\boldsymbol{z},n)}{p^{\pi}(\boldsymbol{s}|\boldsymbol{z},n)p^{\pi}(\eta|\boldsymbol{z},n)} - 1\right)^2 p^{\pi}(\boldsymbol{s}|n)\mathrm{d}\boldsymbol{s}\mathrm{d}\eta,\\
&= \frac{1}{N}\sum_{n=1}^{N}\iint \frac{p^{\pi}(\eta,\boldsymbol{z}|n)^2}{p^{\pi}(\eta|n)p^{\pi}(\boldsymbol{z}|n)^2}\left(\frac{p^{\pi}(\eta,\boldsymbol{s}|\boldsymbol{z},n)}{p^{\pi}(\boldsymbol{s}|\boldsymbol{z},n)p^{\pi}(\eta|\boldsymbol{z},n)} - 1\right)^2 p^{\pi}(\boldsymbol{s}|n)\mathrm{d}\boldsymbol{s}\mathrm{d}\eta,
\end{aligned}
$$

which concludes the proof. (Q.E.D.)

References

1. Sutton, R.S., Barto, G.A.: Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA, USA (1998)
2. Lagoudakis, M.G., Parr, R.: Least-squares policy iteration. Journal of Machine Learning Research **4(Dec)** (2003) 1107–1149
3. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning **8** (1992) 229–256
4. Dayan, P., Hinton, G.E.: Using expectation-maximization for reinforcement learning. Neural Computation **9**(2) (1997) 271–278
5. Kakade, S.: A natural policy gradient. In: Advances in Neural Information Processing Systems 14. (2002) 1531–1538
6. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research **3**(Mar) (2003) 1157–1182
7. Keller, P.W., Mannor, S., Precup, D.: Automatic basis function construction for approximate dynamic programming and reinforcement learning. In: Proceedings of the 23rd International Conference on Machine learning. (2006) 449–456
8. Parr, R., Painter, C.W., Li, L., Littman, L.M.: Analyzing feature generation for value-function approximation. In: Proceedings of the 24th International Conference on Machine Learning. (2007) 737–744
9. Parr, R., Li, L., Taylor, G., Painter, C.W., Littman, L.M.: An analysis of linear models, linear value-function approximation, and feature selection for reinforcement learning. In: Proceedings of the 25th International Conference on Machine Learning. (2008) 752–759

10. Goldberger, J., Roweis, S., Hinton, G., Salakhutdinov, R.: Neighbourhood components analysis. In Saul, L.K., Weiss, Y., Bottou, L., eds.: Advances in Neural Information Processing Systems 17, Cambridge, MA, USA, MIT Press (2005) 513–520
11. Morimoto, J., Hyon, S., Atkeson, C.G., Cheng, G.: Low-dimensional feature extraction for humaniod locomotion using kernel dimension reduction. In: Proceedings of 2007 IEEE International Conference on Robotics and Automation. (2008) 2711–2716
12. Kroon, M., Whiteson, S.: Automatic feature selection for model-based reinforcement learning in factored mdps. In: Proceedings of the 2009 International Conference on Machine Learning and Applications. (2009) 324–330
13. Fukumizu, K., Bach, F.R., Jordan, M.I.: Kernel dimension reduction in regression. Annals of Statistics **37**(4) (2009) 1871–1905
14. MacKay, D.J.C.: Information Theory, Inference, and Learning Algorithms. Cambridge University Press, Cambridge, UK (2003)
15. Suzuki, T., Sugiyama, M., Kanamori, T., Sese, J.: Mutual information estimation reveals global associations between stimuli and biological processes. BMC Bioinformatics **10**(1) (2009) S52
16. Vapnik, V.N.: Statistical Learning Theory. Wiley, New York, NY, USA (1998)
17. Kanamori, T., Hido, S., Sugiyama, M.: A least-squares approach to direct importance estimation. Journal of Machine Learning Research **10** (Jul. 2009) 1391–1445
18. Kanamori, T., Suzuki, T., Sugiyama, M.: Condition number analysis of kernel-based density ratio estimation. Technical report, arXiv (2009) http://www.citebase.org/abstract?id=oai:arXiv.org:0912.2800.
19. Song, L., Smola, A., Gretton, A., Borgwardt, K., Bedo, J.: Supervised feature selection via dependence estimation. In: Proceedings of the 24th International Conference on Machine Learning. (2007) 823–830