A Test for Suboptimal Actions in Markovian Decision Problems

J. MacQueen


Stable URL:
http://links.jstor.org/sici?sici=0030-364X%28196705%2F06%2915%3A3%5C559%3AATFSAI%3E2.0.CO%3B2-K

*Operations Research* is currently published by INFORMS.

Your use of the JSTOR archive indicates your acceptance of JSTOR’s Terms and Conditions of Use, available at http://www.jstor.org/about/terms.html. JSTOR’s Terms and Conditions of Use provides, in part, that unless you have obtained prior permission, you may not download an entire issue of a journal or multiple copies of articles, and you may use content in the JSTOR archive only for your personal, non-commercial use.

Please contact the publisher regarding any further use of this work. Publisher contact information may be obtained at http://www.jstor.org/journals/informs.html.

Each copy of any part of a JSTOR transmission must contain the same copyright notice that appears on the screen or printed page of such transmission.

JSTOR is an independent not-for-profit organization dedicated to creating and preserving a digital archive of scholarly journals. For more information regarding JSTOR, please contact support@jstor.org.
A TEST FOR SUBOPTIMAL ACTIONS IN MARKOVIAN DECISION PROBLEMS

J. MacQueen

Stanford University, Stanford, California

(Received August 9, 1966)

In a Markovian decision problem, choice of an action determines an immediate return and the probability of moving to the next state. It is desired to maximize the expected total of discounted future returns. If upper and lower bounds on the optimal expected return are available, a simple test is described that may show that certain actions are suboptimal, permanently eliminating them from further consideration. This test may be incorporated into the dynamic programming routine for solving the decision problem. This was tried on Howard's automobile replacement problem, using the upper and lower bounds described in "A Modified Dynamic Programming Method" (J. Math. Anal. and Appl. 14, April, 1966). The amount of computation required by the dynamic programming routine was reduced, conservatively, by 75 per cent.

In a typical Markovian decision problem, $X_1, X_2, \ldots$ is a sequence of random variables taking values in a finite set of states $S$, and controlled by a decision maker who at each time $t = 1, 2, \ldots$ observes the state $X_t$ at that time and picks an action $a_t$ belonging to a fixed finite set $A$. This determines two things, an amount of income $g(X_t, a_t)$ that is received immediately, and the probability of moving to the next state $X_{t+1}$, viz., $Pr[X_{t+1} = y | X_t = x, a_t = a]$, and any past up through $t-1] = p(y; x, a)$. The functions $g$ and $p$ are known. Future income is discounted using the factor $\alpha$, $0 < \alpha < 1$, so that the discounted return is

$$g(X_1, a_1) + \alpha g(X_2, a_2) + \alpha^2 g(X_3, a_3) + \cdots$$

A policy $r$ is a rule that determines each of the actions $a_t$, $t = 1, 2, \ldots$ as a function of the sequences $X_1, X_2, \ldots, X_t$ and (possibly) $a_1, a_2, \ldots, a_{t-1}$. Given an initial state, it is desired to choose a policy so as to maximize the expected discounted return. Let $u_r(x)$ be the expected discounted return using policy $r$ and given $X_1 = x$, and let $u^*(x) = \sup_r u_r(x)$. It is known that $u^*$ can be achieved by a stationary policy.[1,2]

The purpose of this note is to point out that if we are given functions $u'$ and $u''$ on $S$, which are, respectively, lower and upper bounds for $u^*$, it may be possible to eliminate immediately certain actions as being suboptimal. The test is: The action $a$ is suboptimal when $x$ is the state, if

$$g(x, a) + \alpha \sum_y u''(y)p(y; x, a) < \max_a [g(x, a) + \alpha \sum_y u'(y)p(y; x, a)].$$  \hfill (1)

This is immediate upon observing that

$$g(x, a) + \alpha \sum_y u^*(y)p(y; x, a) \leq g(x, a) + \alpha \sum_y u''(y)p(y; x, a),$$  \hfill (2)

† Visiting Associate Professor of Statistics.
since \( u^*(y) \) is an upper bound on \( u^* \), while

\[
\max_x [g(x, a) + \alpha \sum_y u'(y)p(y; x, a)] \leq \max_x [g(x, a) + \alpha \sum_y u^*(y)p(y; x, a)],
\]

(3) since \( u' \) is a lower bound. From (1), (2), and (3), we conclude

\[
g(x, a) + \alpha \sum_y u^*(y)p(y; x, a) < \max_x [g(x, a) + \alpha \sum_y u^*(y)p(y; x, a)].
\]

(4) Thus we would never use action \( a \) under the optimal policy when in state \( x \), since we can do better by taking the maximizing action on the right in (4), and proceeding optimally thereafter. The right side of (4) will of course equal \( u^*(x) \).

To illustrate the potential usefulness of this simple test, the automobile replacement problem described by Howard (reference 1, p. 98), was tackled by means of the (slightly) modified dynamic programming method described in reference 4.

<table>
<thead>
<tr>
<th>Iteration:</th>
<th>2</th>
<th>5</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>37(a)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum number eliminated:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td>31</td>
<td>34</td>
<td>37</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>Mean number eliminated:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.9</td>
<td>20.8</td>
<td>34.1</td>
<td>37.5</td>
<td>38.9</td>
<td>40.5</td>
<td></td>
</tr>
<tr>
<td>Per cent eliminated:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.5</td>
<td>50.7</td>
<td>83.1</td>
<td>91.4</td>
<td>94.8</td>
<td>98.7</td>
<td></td>
</tr>
<tr>
<td>Number of states for which the optimal action is known:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>8</td>
<td>17</td>
<td>20</td>
<td>23</td>
<td></td>
</tr>
</tbody>
</table>

(a) This number represents the end of a five-minute computer run (IBM 7094), which happened to be the amount of time that was requested. Computation facilities were provided by the Western Data Processing Center.

This method yields sequences of lower and upper bounds, \( \{u_n'\} \) and \( \{u_n''\} \), which converge monotonically from below and above respectively, to \( u^* \). These bounds are defined as follows: First, let \( v_1 \) by any convenient function with \( v_1(s) = 0 \) for a selected state \( s \), and define the sequence \( v_1, v_2, \ldots \) by

\[
v_{n+1}(x) = \max_x [g(x, a) + \alpha \sum_y v_n(y)p(y; x, a)]
\]

\[
- \max_x [g(s, a) + \alpha \sum_y v_n(y)p(y; s, a)],
\]

(5) so that \( v_n(s) = 0 \), \( n = 1, 2, \ldots \). Then

\[
u_n' = v_n + L_n'(1-\alpha)^{-1},
\]

\[
u_n'' = v_n + L_n''(1-\alpha)^{-1},
\]

(6) where the constants \( L_n' \) and \( L_n'' \) are given by

\[
L_n' = \min_x \{ \max_y [g(x, a) + \alpha \sum_y v_n(y)p(y; x, a)] - v_n(x) \},
\]

\[
L_n'' = \max_x \{ \max_y [g(x, a) + \alpha \sum_y v_n(y)p(y; x, a)] - v_n(x) \}.
\]

(7) Proofs of the monotonicity and convergence to \( u^* \) are in reference 4. Note that if
\( v_1 = 0 \), the sequence of policies corresponding to the maximizing actions on the right-hand side in (5) is the same sequence produced by standard dynamic programming.

Using the bounds (6), the suboptimality test was applied to each action and state at each iteration \( n \) (this is possible at little additional computational cost), and the number of actions thus eliminated from consideration at each state once and for all was counted. In the automobile replacement problem there are essentially 42 possible actions from each of the 41 possible states. The results after 2, 5, 10, 20, 30, and 37 iterations are shown in Table 1. The entries give the minimum, over states, of the number of actions eliminated, the mean number of actions eliminated, both in absolute number and as percentages of 41, and the number of states for which the optimal action was actually determined, all but one action from that state having been eliminated.

At some cost in computer memory it would be possible in applying the dynamic programming method, not to evaluate at each iteration the actions eliminated by this test. This would save, for each subsequent iteration, for each action eliminated, the computation of 
\[
g(x, a) + \alpha \sum_y p(y; x, a) \hat{u}(y)
\]
where \( \hat{u} \) is the current estimate of \( u^* \). In the automobile replacement problem, assuming 30 iterations were performed, about 75 per cent of the total computation would have been saved. Thirty iterations represents a fair place to stop, since at this point the upper and lower bounds differed by less than 2 per cent at every state, and hence it could be asserted that their mean differed by less than 1 per cent from the optimal value. It turned out that in fact the mean of the upper and lower bounds differed from \( u^* \) by at most 0.08 per cent at this point. It would thus appear that by combining the test with the dynamic programming method, a quite efficient technique might result. The freedom from error control problems has been noted in reference 4.

REFERENCES