

Рогова О.А. Об одном подходе к репрезентативности обучающегося алгоритма. // Проблемы информатики в образовании, управлении, экономике и технике: Сб. статей IX Междунар. научно-техн. конф. – Пенза: ПДЗ, 2009. – С. 103-106.

## ОБ ОДНОМ ПОДХОДЕ К РЕПРЕЗЕНТАТИВНОСТИ ОБУЧАЮЩЕГОСЯ АЛГОРИТМА

О.А. Рогова

Тольяттинский государственный университет,  
г. Тольятти, Россия

В данной работе рассматривается репрезентативность обучающихся алгоритмов, максимизирующих математическое ожидание  $I$  на отрезке  $T$ . Проверяется соответствие полученных алгоритмов конкретным характеристикам, которое позволяет говорить о репрезентативности обучающихся алгоритмов.

### **Rogova O.A. On an approach to the representativeness of learned algorithm.**

The article runs about the representativeness of learned algorithms, which maximize the expectation value  $I$  on the segment  $T$ . The correspondence between the received algorithms and concrete characteristics which makes possible to speak on the representativeness of learned algorithms is checked.

Корректная постановка задачи возможна, если предположить наличие априорной информации в виде множества  $\Omega$  возможных  $F$  и априорного распределения для элементов этого множества. Пусть  $I_{A\xi}$  – математическое ожидание  $I$  для алгоритма  $A$  при априорном распределении  $\xi$ :

$$I_{A\xi} = \int_{\Omega} I_A d\xi.$$

При анализе оптимальности обучения используются следующие определения:

1. Алгоритм  $A_\delta$  –  $\delta$ -оптимальный, если обеспечивает максимум  $I$  при неизвестном заранее  $F$ . Пусть  $M = \{A_\delta\}$  – набор  $\delta$ -оптимальных алгоритмов для множества возможных  $F$ ;  $J_\xi = \{I_{A_\delta}\}$  – набор соответствующих значений  $I$ . Система оптимальна, когда  $I$  достигает максимума. Обозначим:  $I_\xi^* = \sup(J_\xi)$ , где  $I_\xi^*$  – наилучшее значение качества управления, получаемое при соответствующем выборе  $\delta$ -оптимального алгоритма и при априорном распределении  $\xi$ .

2. Алгоритм  $A_0$  – обучающийся, если  $I_{A_0\xi} > I_\xi^*$  (обучающаяся система должна быть лучше, чем наилучшая из необучающихся систем).

3. Алгоритм  $A$  – оптимально обучающийся, если  $I_{A^*\xi} > I_{A\xi}$  для любого  $A$  (это позволяет сравнить разные обучающиеся системы между собой) [1].

Рассмотрим задачу оптимального управления. Пусть время дискретно и разбито на такты:  $0, 1, 2, \dots, T$ ;  $n$  – номер такта;  $T$  – полное время управления. В каждом такте выбирается одно из двух управлений  $u_1$  и  $u_2$ . Объект управления характеризуется вероятностями  $p_1$  и  $p_2$  получения значений  $z = 1$  и  $z = 0$  при выборе соответствующих управлений:

$$p(z = 1/u_1) = p_1; p(z = 1/u_2) = p_2; p(z = 0/u_1) = 1-p_1; p(z = 0/u_2) = 1-p_2;$$
$$I = \sum_{n=0}^T z_n.$$

Значения вероятностей  $p_1$  и  $p_2$  неизвестны, для них задается априорное распределение, например, равномерное для значений  $0 \leq p_1 \leq 1$ ;  $0 \leq p_2 \leq 1$ . Нужно найти алгоритм, максимизирующий математическое ожидание  $I$  на отрезке  $T$ .

Алгоритм  $A_\delta$  для каждой пары возможных значений  $p_1$  и  $p_2$  заключается в выборе того управления  $u$ , для которого  $p(z = I)$  максимально. Алгоритм управления  $A$  должен соответствовать характеристикам:

1. Алгоритм  $A_\delta$  –  $\delta$ -оптимальный, если обеспечивает максимум  $I$  при неизвестном заранее  $F$ .
2. Алгоритм  $A_0$  – обучающийся, если  $I_{A_0\xi} > I_\xi^*$ .
3. Алгоритм  $A$  – оптимально обучающийся, если  $I_{A*\xi} > I_{A\xi}$  для любого  $A$  [1].

Часто утверждается, что в возникающих на практике задачах входные данные поступают в соответствии с некоторым вероятностным распределением, и поэтому сложность «в наихудшем случае» в таких случаях неадекватна. В настоящее время многими учеными ведется работа в направлении построения теории, отдельно описывающей алгоритмы для предположительно правильного распределения в каждом конкретном случае, которые достаточно адекватно описывали бы многие практические ситуации. Наибольший прогресс был достигнут в работах американских теоретиков Л. Левина [2] и Ю. Гуревича [3]. В данной работе подход к репрезентативности был сделан в этом же смысле.

Выборочное наблюдение используют для проверки результатов сплошного наблюдения. Качество результатов выборочного наблюдения зависит от того, насколько состав выборки представляет генеральную совокупность. Чтобы обеспечить репрезентативность выборки, необходимо соблюдать принцип случайности отбора единиц: на включение или исключение объекта из выборки не может повлиять какой-либо иной фактор, кроме случая [4]. В работе при проведении компьютерного эксперимента для построения последовательности случайных величин  $I$  использовалась стандартная функция языка программирования C++ `srand(time(NULL))`. Случайность отбора единиц данной функцией проверялась по статистическим критериям, описанным в [5]. Соответствие выборки этим критериям позволяет говорить о репрезентативности выборки.

Алгоритмы, соответствующие выбранным случайным величинам  $I$ , проверяются на соответствие рассматриваемым характеристикам. На основании этого выносится решение о том, что вся сгенерированная последовательность будет соответствовать выбранным характеристикам. Программная реализация показала, что обучающиеся алгоритмы, максимизирующие математическое ожидание  $I$  на отрезке  $T$ , соответствуют рассматриваемым характеристикам, а это позволяет говорить о репрезентативности обучающихся алгоритмов.

#### Библиографический список

1. Пивнева С.В. Математическое моделирование оптимального управления образовательной средой // Татищевские чтения: Актуальные проблемы науки и практики : материалы Междунар. науч. конф. – Тольятти, 2004.
2. Levin L. Randomness and Nondeterminism. – <http://www.cs.bu.edu/fac/lnd/expoicm94.htm>
3. Gurevich Y., Veanes M., Wallace C. Can Abstract State Machines Be Useful in Language Theory? Theoretical Computer Science, 376, 2007.

4. Чернова Т.В. Экономическая статистика: Статистическое наблюдение. – [www.aup.ru/books/m81/2.htm](http://www.aup.ru/books/m81/2.htm)
5. Кнут Д. Искусство программирования. Т. 1, 2.