

На более высоких уровнях иерархии оценивание успеваемости производится путем композиции нечетких функций принадлежности

$$\mu_{\text{тек}} = \omega_{\text{лек}}\mu_{\text{лек}} + \omega_{\text{прак}}\mu_{\text{прак}} + \dots, \quad (2)$$

где ω - весовой коэффициент концепта, заданный при проектировании предметной области, μ - характеристическая функция каждого концепта.

Характеристические функции также задаются преподавателем в момент проектирования предметной области и зависят от типа концепта – лекции, лабораторные работы, практические, тесты.

Данный подход предоставляет большие возможности для построения гибкой иерархической системы оценивания усвоения учебного материала, учитывая взаимосвязи на различных этапах обучения. Использование характеристической функции для оценивания уровня знаний позволяет сделать менее категоричным подход к определению знаний студента, а также предоставить возможность преподавателю изменять подход к оцениванию каждого концепта.

УДК 681.3

Декомпозиция логической системы на части по сложным условиям неопределенности

Прихожий А.А.

Белорусский национальный технический университет

Не полностью определенная логическая система [1] описывается выражениями вида (fd) , где f – логическая (булева) функция, зависящая от переменных x_1, \dots, x_n ; d – характеристическая булева функция области определенности, также зависящая от x_1, \dots, x_n . Инверсия $\neg d$ функции описывает область неопределенности. Если $d=x$, то выражение (fd) упрощается до выражения $(f_{x=1}|x)$, где $f_{x=1}$ – остаточная функция, получаемая при подстановке значения 1 вместо переменной x . Если $d=\neg x$, то выражение (fd) упрощается до выражения $(f_{x=0}|\neg x)$, где $f_{x=0}$ – остаточная функция, получаемая при подстановке значения 0 вместо переменной x . Для двухместных булевых операций, сводящихся к конъюнкции \wedge , и произвольных булевых функций g и h выполняется:

$(f_{g=1, h=1} | g \wedge h)$, где $f_{g=1, h=1}$ – остаточная функция при $g=h=1$;

$(f_{g=1, h=0} | \neg(g \rightarrow h))$, где $\neg(g \rightarrow h) = g \wedge \neg h$ – инверсия импликации;

$(f_{g=0, h=1} | \neg(g \leftarrow h))$, где $\neg(g \leftarrow h) = \neg g \wedge h$ – инверсия обратной импликации;

$(f_{g=0, h=0} | g \downarrow h)$, где $g \downarrow h = \neg g \wedge \neg h$ – стрелка Пирса.

Для следующих четырех двухместных булевых операций, используемых в качестве d и сводящихся к дизъюнкции \vee , имеют место следующие декомпозиции:

$$(f|g\vee h) = g\&(f|g) + \sim g\&(f|g\wedge h);$$

$(f|g\leftarrow h) = g\&(f|g) + \sim g\&(f|g\wedge \sim h)$, где $g\leftarrow h = g\vee \sim h$ – обратная импликация;

$$(f|g\rightarrow h) = \sim g\&(f|\sim g) + g\&(f|g\wedge h), \text{ где } g\rightarrow h = \sim g\vee h \text{ – импликация;}$$

$$(f|g/h) = \sim g\&(f|\sim g) + g\&(f|g\wedge \sim h), \text{ где } g/h = \sim g\vee \sim h \text{ – штрих Шеффера.}$$

Здесь ‘ \sim ’ – частичная инверсия; ‘ $\&$ ’ – частичная конъюнкция; ‘+’ – частичная дизъюнкция [1]. Примечательно, что правые части разложений используют для описания области определенности d только конъюнкцию. Все продукты разложений проще исходной функции f . Для более сложной характеристической функции $d = c\wedge g\vee \sim c\wedge h$ выполняется разложение:

$$(f|c\wedge g\vee \sim c\wedge h) = c\&(f|c\wedge g) + \sim c\&(f|\sim c\wedge h).$$

Его частным случаем является разложение по операции исключающее или:

$$(f|c\oplus g) = c\&(f|c\wedge \sim g) + \sim c\&(f|\sim c\wedge g).$$

Литература:

1. Прихожий А.А. Частично определенные логические системы и алгоритмы // Минск: БНТУ. – 2013. – 343 с.

УДК 681.3

Генетический алгоритм разбиения коллектива программистов на группы

Прихожий А.А., Ждановский А.М.

Белорусский национальный технический университет

Для оптимизации процессов организации и управления человеческими ресурсами предложено следующее решение [1], реализованное с помощью генетического алгоритма и основанное на распределении программистов по рабочим группам. Распределение учитывает несколько важнейших факторов: производительность программистов, повышение или понижение производительности при их совместной работе, дополнительные затраты на интерфейсы между программистами и между рабочими группами.

Пусть дано множество программистов $P = \{p_1, \dots, p_n\}$ и общее время T работы над проектом. Задача заключается в разбиении множества программистов на множество групп $G = \{g_1, \dots, g_m\}$. Производительность труда определяется величиной r_p , равной количеству условных человеко-