

# ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И ЭКСПЕРТНЫЕ СИСТЕМЫ

(Вычислительные системы)

1997 года

Выпуск 160

УДК 001.18:[001+621/681].001/009(063)

## САМООБУЧАЮЩИЙСЯ ГЕНЕТИЧЕСКИЙ АЛГОРИТМ ПРОГНОЗИРОВАНИЯ (LGAP)<sup>1</sup>

Н.Г. Загоруйко

### В в е д е н и е

Традиционно алгоритмы самообучения анализируют протоколы, в которых имеются описания входных воздействий и реакций изучаемой системы на эти воздействия. Раньше в кибернетической литературе задачи установления зависимостей между характеристиками входа и выхода назывались задачами анализа "черного ящика". Однако, имеются еще более сложные ситуации, когда входные воздействия в протоколе не отражены и проявляют себя лишь косвенно, через изменения выходных характеристик наблюдаемых объектов. Дальнейшим осложнением является наличие скрытых внутренних влияний характеристик одних объектов на характеристики других объектов. Такими свойствами обладают, например, протоколы наблюдений за характеристиками людей, взаимодействующих друг с другом в процессе решения неких внешних задач, не отраженных в протоколе. Другим примером являются протоколы курсов валют или ценных бумаг на бирже, которые зависят от неизвестных внешних событий.

---

<sup>1</sup>Работа выполнена при поддержке гранта РФФИ (№ 97-06-80312) и в рамках проекта, поддержанного грантом Госкомитета по высшему образованию РФ.

В таких ситуациях "абсолютно черного ящика" задача предсказания будущего состояния характеристик наблюдаемых объектов может быть решена лишь на базе двух гипотез: гипотезы "повторяемости" (в прошлом встречались такие же или аналогичные внешние воздействия) и гипотезы "адекватности реакции" (похожие воздействия вызывают похожие реакции).

В данной работе описывается адаптивный генетический алгоритм LGAP (Learning Genetic Algorithm for Prognosis) для извлечения закономерностей (знаний) из такого рода данных и использования этих знаний для прогнозирования будущих событий. Алгоритм LGAP существенно использует идеи алгоритма ZET для заполнения пробелов в эмпирических таблицах [1,2]. В связи с этим напомним основное содержание алгоритма ZET.

### 1. Алгоритм ZET для заполнения пробелов

Для простоты будем считать, что исходные данные имеют вид таблицы типа "объект-свойство", строки в которой  $(1, 2, \dots, i, \dots, l, \dots, m)$  соответствуют  $m$  объектам  $a(i)$ , а столбцы  $(1, 2, \dots, j, \dots, k, \dots, n)$  —  $n$  признакам или свойствам  $x(j)$ . Предположим, что значение  $b(ij)$   $j$ -го признака  $i$ -го объекта нам не известно, и мы хотим получить его прогнозируемое значение. Предсказания такого рода основываются на следующем ряде эвристических гипотез:

Н1. Объекты, похожие по  $k$  свойствам, обычно похожи и по  $(k + 1)$ -му свойству.

Н2. Эмпирические таблицы с реальными данными обычно содержат избыточность: в них имеются объекты, похожие друг на друга, есть и признаки, зависящие друг от друга.

Н3. Из всех видов имеющихся в таблице зависимостей достаточно использовать самые простые — линейные зависимости.

Н4. Для предсказания значения конкретного элемента таблицы нецелесообразно использовать ее всю.

Лучше использовать специально выбранную часть таблицы, так называемую "компетентную" подматрицу.

Элементы, стоящие в строке  $i$  и столбце  $j$ , на пересечении которых находится предсказываемый элемент, будем называть "базовыми". После нормировки значений каждого столбца таблицы выбирается компетентная подматрица, в которую попадают  $r$  строк, наиболее похожих на базовую строку  $i$ , и  $v$  столбцов, наиболее тесно связанных с базовым столбцом  $j$ . Мерой близости строк может служить расстояние между ними в евклидовом пространстве признаков, а мерой связанности (зависимости) признаков — модуль коэффициента корреляции. Каждой  $l$ -й строке компетентной подматрицы приписывается вес  $L(il)$ , пропорциональный ее компетентности, которая тем выше, чем меньше расстояние между  $i$ -й и  $l$ -й строками, и чем больше размерность признакового пространства, в котором определялось это расстояние. Точно также, компетентность  $L(jk)$  каждого  $k$ -го столбца тем больше, чем больше его корреляция с  $j$ -м столбцом и чем больше строк участвовало в вычислении корреляции. Величина  $L$  меняется от 0 до 1.

Далее строится  $r$  линейных регрессий между  $i$ -й строкой и всеми  $r$  компетентными строками. По каждой такой регрессии между  $i$ -й и  $l$ -й строками, зная значение элемента  $b(lj)$  в  $l$ -й строке, можно получить вариант предсказания значения пропущенного элемента в  $i$ -й строке —  $b(ij, l)$ . Усреднение полученных вариантов с весами компетентности дает вариант, выработанный с участием всех  $r$  строк:

$$b^*(ijl) = \frac{\sum_{l=1}^r b(ij, l) * L(il)^\alpha}{\sum L(il)^\alpha}.$$

Показатель степени  $\alpha$  отражает отношение к коэффициентам компетентности. При  $\alpha = 0$   $L = 1$ , и все подсказки считаются одинаково важными, а при больших  $\alpha$  учитываются только подсказки от самых компетентных строк. Значения выбираются для каждой запол-

няемой клеточки таблицы отдельно. Для этого все известные базовые элементы  $i$ -й строки предсказываются с помощью этого же алгоритма при разных значениях  $\alpha$ . Сравнение истинных и предсказанных величин позволяет выбрать то значение  $\alpha$ , при котором ошибка предсказания была минимальной, а заодно и оценить ожидаемую ошибку предсказания неизвестного элемента.

Аналогичным образом определяется и вариант, полученный при усреднении подсказок  $b(ij, k)$  от всех  $v$  столбцов с весом, равным компетентности  $L(jk)$  этих столбцов:

$$b^*(ij, k) = \frac{\sum_{k=1}^v b(ij, k) * L(jk)^\alpha}{\sum L(jk)^\alpha}.$$

Окончательный вариант предсказываемого значения получается, например, путем простого усреднения подсказок от строк и от столбцов:

$$b^*(ij) = \frac{\{b^*(ij, l) + b^*(ij, k)\}}{2}.$$

## 2. Обучающийся генетический алгоритм прогнозирования LGAP.

В алгоритме LGAP выделяется три этапа его работы: формирование базовых элементов (базовых "штаммов"), поиск компетентных штаммов и предсказание заданного неизвестного элемента.

2.1. I-й этап: Формирование множества базовых штаммов. Вместо трехвходовой таблицы "объект-свойство-время", обычно рассматриваемой в реальных задачах, будем пояснять алгоритм на примере двухвходовой таблицы "объект-время". Каждый элемент  $a(i, t)$  таблицы (протокола событий за  $T$  прошедших дней) отражает значение одной характеристики  $i$ -го объекта ( $i = 1, 2, \dots, Y$ ) в  $t$ -й момент времени. Моменты времени ( $t = 1, 2, \dots, T$ ) упорядочены в ней по "возрасту": самые свежие данные имеют индекс  $t = 1$ , данные за предшествующий день —  $t = 2$  и т.д. до дня с индексом  $t = T$ .

В алгоритме ZET компетентность элементов оценивалась по их схожести на некоторое "базовое" подмножество элементов, находящихся в одной и той же строке или в одном и том же столбце. В данном алгоритме среди  $G(g = 1, 2, 3, \dots, G)$  элементов, входящих в состав базового множества (будем называть такие множества штаммами мощности  $G$ ) для прогнозирования элемента  $a(i, 0)$ , могут оказаться любые элементы, взятые из протокола за  $\tau$  последних дней, т.е. элементы с произвольными значениями вектора их координат  $v(i, t)$ ,  $t < \tau$ .

Выбор базовых штаммов делается среди  $N$  элементов таблицы, где  $N = Y * \tau$ . Ясно, что при количестве элементов в штамме  $G$ , число вариантов базовых штаммов будет равно числу сочетаний из  $N$  по  $G$ . Нетрудно видеть, что трудоемкость алгоритма даже при небольших значениях  $N$  и  $G$  может превысить возможности компьютера рядового пользователя. Для сокращения перебора будет использоваться комбинация метода последовательного наращивания числа элементов, предложенного Барабашем Ю.Л. [3], с методами генетического программирования.

Метод Барабаша применительно к нашей задаче может выглядеть так. Вначале оценивается ожидаемая ошибка прогноза элементов  $a(i, 0)$  при использовании в качестве базовых всех  $N$  элементов по одному ( $G = 1$ ). Выбирается подмножество  $n_1$  наилучших вариантов ( $n_1 < N$ ). Затем просматриваются базовые штаммы из пар, образованных каждым из  $n_1$  элементов, отобранных на первом шаге, и всеми остальными  $N - 1$  элементами. Таких вариантов будет  $n_1 * (N - n_1)$ . Из них выбираем  $n_2$  лучших пар, к которым по очереди добавляются по одному остальные  $N - 2$  элемента (здесь число вариантов около  $n_2 * (N - n_2)$ ) и тем же путем отбирается  $n_3$  лучших троек и т.д. до заданного количества элементов  $G$  в базовом штамме. При постоянной величине выбора на каждом шаге  $n$  лучших вариантов общее число ( $B$ ) просматриваемых вариантов будет порядка  $G * n * N$ , что, на-

пример, при  $G = 12$ ,  $n = 2$  и  $N = 36$  меньше объема полного перебора в миллион раз.

Описанный процесс можно считать некоторой моделью, сочетающей мутации (в виде добавления новых элементов у потомков) с естественным отбором лучших из них для использования в качестве "родительских" штаммов в следующем поколении.

Однако, известно, что этот метод обеспечивает получение лишь локально-оптимального решения. Попытаемся улучшить найденные решения. Применим на каждом шаге (т.е. при каждом новом значении  $G$ ) процедуру "скрещивания". Для этого упорядочим элементы всех отобранных  $nG$  базовых штаммов по индексу времени. Если два элемента имеют одинаковый индекс  $i$ , то упорядочим эти элементы по номеру объекта  $i$ . В соответствии с этим порядком присвоим всем элементам базового штамма номера от 1 до  $G$ .

Из каждой пары родителей можно сформировать большое количество потомков путем замены  $k$  элементов первого родителя на  $k$  элементов второго родителя с одинаковыми порядковыми номерами этих элементов. При каждом  $k$  число новых потомков будет равно  $C_G^k$ . При  $k > G/2$  будут появляться дублиеры уже имеющихся потомков, поэтому общее возможное число различных потомков равно  $\sum_{k=1}^{G/2} C_G^k$ . Можно ограничиться меньшим числом потомков, например, двумя, которые образованы соединением первой половины одного родителя со второй половиной второго родителя (и наоборот). Качество таких гибридных штаммов оценивается наряду с качеством породивших их родителей и из этого набора в  $2 * nG$  вариантов отбирается  $nG$  наилучших штаммов для последующего участия в мутациях и естественном отборе.

Упомянутая постоянно процедура оценки качества базовых штаммов включает в себя этап поиска компетентных штаммов и этап оценки ожидаемой ошибки предсказания с использованием этих штаммов. Перей-

дем к описанию процедуры отбора множества компетентных штаммов.

**2.2. II-й этап: Отбор компетентных штаммов.** Элементы базового штамма в массиве исходных данных помечены конкретными значениями индексов их координатных векторов  $v(i, t)$ . Набор этих векторов  $v_1(i, t), v_2(i, t), \dots, v_g(i, t), \dots, v_G(i, t)$  описывает структуру (или архитектуру) конкретного штамма, состоящего из  $G$  элементов. Если индекс  $t$  у всех элементов данного штамма увеличить на заданное число  $q$ , то мы получим штамм той же структуры, что и исходный, но только сдвинутый во времени на  $q$  шагов назад. Назовем такой штамм "изоморфным" данному базовому штамму.

Если через  $t_{\max}$  обозначить момент времени, наибольший среди всех элементов базового штамма, то в таблице содержится  $(T - t_{\max})$  изоморфных ему штаммов. Мера "похожести" между изоморфными штаммами будем оценивать, например, через коэффициент их корреляции  $kor$ .

Компетентными будем считать такие изоморфные штаммы, модуль коэффициента корреляции которых с базовым штаммом превышает некоторый заданный порог, например,  $kor = 0,85$ . Можно поступать и по другому: включать в состав компетентных заданное число  $d$  изоморфных штаммов с наибольшими значениями  $kor$ .

Таким способом для каждого заданного базового штамма выбирается множество его компетентных аналогов.

**2.3. III этап: Прогнозирование элемента  $a(i, 0)$ .** Добавим к базовому штамму элемент  $a(i, 0)$ , значение которого неизвестно. Получается базовый штамм новой структуры, сдвиг которого на  $q$  шагов по времени выделяет изоморфный ему штамм с элементом  $a(i, q - 1)$ . Строится линейная регрессия между  $G$  элементами базового и изоморфного штаммов. Подстановка элемента  $a(i, q - 1)$  в уравнение этой регрессии позволяет получить величину  $a_q(i, 0)$  в качестве  $q$ -го варианта прогноза для элемента  $a(i, 0)$ . Аналогично получают и все другие варианты прогноза для данного базового штамма:  $a_1(i, 1), a_2(i, 2), \dots$

$\dots, a_q(i, q), \dots, a_d(i, d)$ . Назовем эти элементы предикторами.

Вспомним гипотезу  $H_1$ , используемую в распознавании образов и в алгоритме ZET: "объекты, похожие по  $k$  характеристикам, обычно похожи и по  $(k+1)$ -й характеристике". Так это или нет — в алгоритме ZET выясняется с помощью процедуры подбора коэффициента  $\alpha$ , влияющего на учет компетентности. В данном алгоритме применяется более простая процедура оценки компетентности  $h_s$  предикторов данного базового штамма  $s$ : вычисляется дисперсия  $D_s$  предикторов и по ней принимается решение о степени действительной компетентности  $h_s$ . Большое значение дисперсии означает, что наша гипотеза в данном случае не верна, и компетентность  $D_s$  рассматриваемых предикторов мала.

Если общее число базовых штаммов равно  $S$ , то итоговое значение прогнозируемого элемента  $a(i, 0)$  можно принять равным средневзвешенному значению его предикторов  $a(i, 0) = \frac{\sum a(s) * h_s}{\sum h_s}$  для всех  $s$  от 1 до  $S$ . Здесь  $a(s)$  — среднеарифметическое значение предикторов штамма  $s$ .

По среднему значению компетентности всех  $S$  штаммов можно судить об ожидаемой ошибке прогнозирования: эксперименты показали, что коэффициент корреляции между ошибками прогнозирования и дисперсией предикторов при анализе разных таблиц может колебаться в пределах от 0.2 до 0.7.

В итоге, общая процедура прогнозирования элемента  $a(i, 0)$  состоит из следующих шагов.

1. Формирование базового штамма.
2. Поиск компетентных штаммов среди изоморфных базовому.
3. Формирование множества предикторов.
4. Оценка компетентности базового штамма через дисперсию предикторов.
5. Повторение пп. 1-4  $S$  раз.
6. Получение средневзвешенного значения предсказываемого элемента.

7. Получение оценки ожидаемой ошибки прогнозирования.

Для всех других предсказываемых элементов эта процедура повторяется.

### 3. Трехуровневый отбор предикторов

Более высокая надежность прогноза получается не при описанном выше одноуровневом методе учета компетентности, а при многоуровневой процедуре отбора компетентных предикторов.

Предикторы порождаются разными множествами штаммов. На первом уровне используются компетентные штаммы  $S_k$  из числа изоморфных данному базовому штамму  $S_b$  заданной мощности  $G$ . На этом уровне отбор  $dk$  лучших предикторов делается по величине коэффициента корреляции между  $S_k$  и  $S_b$ .

В результате перебора всех  $r$  базовых штаммов мощности  $G$  получается  $r$  групп предикторов по  $dk$  штук в каждой группе. Отбор лучших групп на этом этапе делается по величине дисперсии значений  $dk$  прогнозов в данной группе. Предпочтение отдается  $dr$  группам с наименьшими дисперсиями прогнозов.

Затем рассматриваются штаммы с другими значениями мощности  $G$ . После просмотра всех  $q$  базовых штаммов различной мощности получается  $q$  множеств предикторов. Отбор лучших  $dq$  среди них делается на третьем этапе по дисперсиям прогнозов, получаемых предикторами, входящими в каждое множество.

На каждом этапе делается взвешенное усреднение значений прогнозов.

### 4. Дисперсионный критерий для выбора стратегии прогноза

В описанных выше процедурах имеется большое число изменяемых параметров — мощность штамма, число элементов таблицы для формирования базовых штаммов, числа предикторов для оценки компетентности и

для усреднения на каждом этапе и т.д. Каждому сочетанию этих параметров соответствует своя стратегия и свои результаты прогноза. Для одного предсказываемого элемента лучшие результаты дает одна стратегия, для другого может оказаться лучшей другая стратегия. Обычно выбор лучшей стратегии делается в режиме ретроспективного прогноза: лучшей считается такая стратегия, применение которой приводило к наименьшей сумме ошибок прогноза известных элементов. Но в ходе таких экспериментов может обнаружиться, что в ряде случаев лучшие результаты давали некоторые другие стратегии. По-видимому, они более точно соответствовали тем закономерностям процесса, которые проявляли себя в данный конкретный момент.

Какую же стратегию использовать для реального прогнозирования неизвестных величин? Где гарантия, что стратегия, чаще других в прошлом приводившая к успеху, будет лучшей и в этом конкретном случае?

Введение описанного выше дисперсионного критерия позволяет дать следующий ответ на этот мучительный для прогнозистов вопрос: применяй все имеющиеся в распоряжении стратегии и отдавай предпочтение той из них, которая дает варианты прогноза, обладающие наименьшей дисперсией.

Эта эмпирическая гипотеза отражает закономерность более высокого уровня (метазакономерность) по сравнению с традиционно используемой гипотезой о том, что хорошие результаты в прошлом обеспечивают хорошие результаты и в будущем.

## 5. Критерии для оценки точности прогноза

Еще один сложный вопрос обычно возникает в процессе прогнозирования: как оценивать то, что мы называем "точностью" прогноза?

Часто берется абсолютное отклонение прогноза  $a'(i, 0)$  от истинного значения  $a(i, 0)$ , деленное на истинное значение:  $d' = \frac{\{a'(i, 0) - a(i, 0)\}}{a(i, 0)}$ .

Такая относительная величина мало чувствительна к ошибкам прогноза больших значений и чрезмерно чувствительна к ошибкам прогноза величин, близких к нулю. Кроме того, разница  $dd$  между минимальным и максимальным значениями различных наблюдаемых характеристик может сильно отличаться друг от друга, и одинаковая относительная ошибка  $d'$  будет приемлемой для принятия решений в одних случаях и неприемлемой в других.

В связи с этим предлагается судить о точности прогноза по величине ошибки, нормированной по разнице  $dd(i)$ : 
$$d = \frac{\{a'(i, 0) - a(i, 0)\}}{dd(i)}.$$

Такая мера обладает одинаковой чувствительностью к ошибкам прогноза для разных значений прогнозируемой характеристики. Ее чувствительность к ошибкам тем выше, чем в меньших пределах колеблется прогнозируемая характеристика, что представляется вполне логичным.

Иногда важно знать не абсолютную величину  $a(i, 0)$  характеристики в будущем, а лишь то, будет ли она больше или меньше значения  $a(i, t)$  в данный момент времени  $t$ . В таких случаях применима мера точности прогноза, учитывающая лишь совпадения знаков:

$$d = 1, \text{ если } \{a(i, 0) > a(i, t)\} \text{ и } \{a'(i, 0) > a(i, t)\} \\ \text{или } \{a(i, 0) < a(i, t)\} \text{ и } \{a'(i, 0) < a(i, t)\}, \\ d = 0.5, \text{ если } \{a(i, 0) = a(i, t)\}, \text{ а } \{a'(i, 0) \neq a(i, t)\} \text{ и} \\ d = 0 \text{ — в других случаях.}$$

## 6. Возможности распараллеливания алгоритма LGAP

Алгоритм LGAP легко распараллеливается на число предсказываемых элементов  $U$ . Каждая из  $U$  линий при заданном числе  $G$  элементов базового множества распараллеливается на  $M^G$  независимых процессов поиска компетентных штаммов и прогнозирования по ним. Обмены между процессами делаются на этапе выбора лучших базовых и создания новых штаммов путем мутаций или

скрещиваний, после чего новые базовые штаммы запускаются в независимые параллельно протекающие процессы.

Так что, машинное время при решении этой задачи на многопроцессорной системе будет почти строго обратно пропорционально числу процессоров в системе.

## 7. Экспериментальная проверка алгоритма LGAP

Экспериментальная проверка алгоритма проводилась на двух таблицах. Одна из них содержала ежедневно публикуемые курсы различных валют на Московской Межбанковской Валютной Бирже за 60 дней ее работы. Проверялась возможность прогнозирования кросс-курсов некоторых из этих валют на один день вперед. Вторая представляла собой таблицу с ежедневными значениями различных обобщенных характеристик, описывающих динамику торгов на Нью-Йоркской фондовой бирже в течение двух лет (1995 и 1996 гг.). Эта таблица опубликована в февральском выпуске американского журнала *Journal of Computational Intelligence in Finance* (файл *Test.csv*)<sup>2</sup> в качестве тестового материала для проверки возможностей прогнозирования значений характеристик на 5 дней вперед. Эти материалы интересны тем, что в них лишь косвенно отражаются сложные внешние процессы, влияющие на формирование курсов валют или на деловую активность участников торгов, и потому задача прогнозирования этих характеристик является заведомо сложной.

Для первого эксперимента были взяты данные за период в 60 дней по 6 валютам (доллар США, немецкая марка, фунт стерлингов, французский франк, японская иена и ЭКЮ). Вычислялись отношения курсов валют друг к другу, и эти кросс-курсы нормировались по времени в диапазоне от 0 до 1. Предсказывались значения кросс-курсов в день  $t$ . В качестве базовых выбирались

---

<sup>2</sup>Материалы можно найти по электронным адресам: 72672.261@compuserve.com или <http://ourworld.compuserve.com/homepages/ftpub/call.htm>.

все неизоморфные штаммы мощностью  $G$  (от 3 до 12), которые можно было построить из элементов  $T$  дней, предшествовавших дню  $t$  ( $T$  от 1 до 10 дней). Компетентные штаммы определялись в оставшемся массиве в  $(60-T-t)$  дней. Кроме  $T$ ,  $t$  и  $G$  в экспериментах варьировались количества компетентных предикторов на всех трех этапах их отбора.

Ввиду того, что после нормировки все кросс-курсы изменяются в одинаковых пределах от 0 до 1, в качестве меры качества прогноза использовалась абсолютная величина ошибки в процентах.

Эксперименты подтвердили целесообразность использования дисперсионного критерия для оценки ожидаемой точности прогноза.

Выяснилось, что наиболее компетентные штаммы концентрируются в непосредственной близости к прогнозируемому дню. Для одного и того же дня при прогнозе разных кросс-курсов оказываются компетентными предикторы, порождаемые разными штаммами, так что настраивать программу на отдельные штаммы, которые в процессе ретроспективного прогнозирования давали наилучшие результаты, нельзя. Нужно использовать все штаммы и для каждого прогнозируемого элемента отбирать группы предикторов по дисперсионному критерию.

Ошибки прогноза для кросс-курсов разных валют бывают разными. Наибольшую трудность представляет прогноз кросс-курсов, в которых задействованы немецкие марки и (или) французские франки.

Для оценки качества получаемых прогнозов иногда используют сравнение с самой простой стратегией прогнозирования, при которой считается, что величина функции и ее знак завтра будут такими же, как и сегодня. При этой стратегии ошибка прогнозирования кросс-курсов в данной таблице равнялась 4.9%, а знаки угадывались правильно в 57.6% случаев.

При наилучшей настройке параметров алгоритма средняя ошибка прогноза равна 4.16%, а знаки изменения кросс-курсов прогнозируются с точностью 64.7%.

Во втором эксперименте делался прогноз одной из характеристик на 5 дней вперед по ее значениям в прошлом. Прогнозировались значения характеристики в каждый из 300 последних дней работы биржи в массиве данных за 1995 и 1996 г.г. (т.е. часть дней конца 1995 г. и все дни 1996 г.). Первичные варианты прогнозов делались с помощью моделей линейной регрессии. К дальнейшему анализу пропускались те варианты, которым соответствовали более высокие значения коэффициентов корреляции между базовым и изоморфными ему штаммами. В окончательном решении участвовали варианты с наименьшими дисперсиями вариантов для каждого дня. Регулируя пороговые значения корреляции и дисперсии, можно получать разную точность прогноза.

За 5 дней анализируемая характеристика изменяется в среднем на 1.446%, а знак изменения сохраняется неизменным в 50% случаев. Средняя ошибка 300 прогнозов, полученных алгоритмом LGAP, оказалась равной 1.432%, а знак изменения характеристики угадывался правильно в 63,7% случаев. Меняя указанные выше пороги, можно выделить дни, в которые ожидаемая ошибка

Количество прогнозов	Ошибка прогноза в %	Количество правильно угаданных знаков	Доля дней с правильно угаданными знаками, в %
300	1.432	191	63.7
297	1.438	189	63.6
242	1.436	161	64.9
239	1.430	157	65.7
229	1.419	152	66.4
226	1.412	151	66.8
...	...	...	...
64	1.145	44	68.8
60	1.156	44	73.3
53	1.102	43	81.1
52	1.133	42	80.8
51	1.118	41	80.4

будет слишком большой и отказаться делать прогнозы на эти дни. Как видно из приведенной таблицы, с увеличением порога по корреляции и уменьшением порога по дисперсии, количество дней, в которые делается прогноз, уменьшается, но в оставшиеся дни точность прогноза величины и знака более или менее плавно увеличивается. При некоторых значениях порогов прогнозы будут выдаваться, например, в среднем один раз за пять дней работы биржи, но с ошибкой в оценке величины курса и знака его изменения на 20% меньшей, чем при ежедневном прогнозе.

### З а к л ю ч е н и е

Алгоритм LGAP предполагается использовать для решения прикладных задач, которые традиционно решались нами с помощью алгоритма ZET: прогноз результатов деятельности предприятий, прогноз урожайности сельскохозяйственных культур, прогноз состояния наблюдаемых пациентов и т.д., а также для решения новых задач прогнозирования процессов экономического, экологического, демографического характера, возникающих, например, при изучении проблемы устойчивого развития ноосферы.

### Л и т е р а т у р а

1. ЗАГОРУЙКО Н.Г., ЕЛКИНА В.Н., ТИМЕРКАЕВ В.С. Алгоритм заполнения пропусков в эмпирических таблицах (алгоритм ZET) // Новосибирск, 1975. — Вып. 61: Вычислительные системы. — С.3-27.
2. Пакет прикладных программ ОТЭКС. /Н.Г.Загоруйко, В.Н.Елкина, С.В.Емельянов, Г.С.Лбов. — М.: Финансы и статистика, 1986. — 160 с.
3. БАРАБАШ Ю.Л., ВАРСКИЙ Б.В. и др. Автоматическое распознавание образов.— Киев: Изд-во КВАИУ, 1963.

Поступила в редакцию  
22 октября 1997 года