ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ

Научный журнал Отделения нанотехнологий и информационных технологий Российской академии наук

Издается с 2007 года
Журнал выходит ежеквартально

Учредители:
Российская академия наук
Институт проблем информатики Российской академии наук

Адрес редакции:

Москва 119333, ул. Вавилова 44, корп. 2, ИПИ РАН,
редакция журнала «Информатика и её применения»
Тел. 8(499)135-86-92, e-mail rust@ipiran.ru

Журнал «Информатика и её применения» включен в «Перечень ведущих рецензируемых научных журналов и изданий, в которых должны быть опубликованы основные научные результаты диссертации на соискание ученой степени доктора и кандидата наук», утвержденный ВАК

Подписной индекс журнала в каталоге «Пресса России» 88018 (годовая подписка)

Информатика и её применения

Том 2 Выпуск 4 Год 2008

СОДЕРЖАНИЕ

Выпускающий редактор *Л. Кокушкина*
Технический редактор *Т. Торжкова*
Художественный редактор *М. Седакова*
Сдано в набор 10.11.08. Подписано в печать 02.12.08. Формат 60 x 84 / 8
Бумага офсетная. Печать офсетная. Усл.-печ. л. 13,0. Уч.-изд. л. 11,8. Тираж 200 экз.
Заказ №
Издательство «ТОРУС ПРЕСС», Москва 121614, ул. Крылатская, 29-1-43
torus@torus-press.ru; http://www.torus-press.ru
Отпечатано в ППП «Типография «Наука» с готовых диапозитивов
Москва 121099, Шубинский пер., д. 6.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2008. Т. 2. Вып. 4. С. 2–11

ДЕЗИНТЕГРИРОВАННАЯ АРХИТЕКТУРА ПАКЕТНОЙ КОММУТАЦИИ

И. А. Соколов[1], В. Б. Егоров[2]

Аннотация: Предложена дезинтегрированная архитектура пакетной коммутации, позволяющая создавать простые и маршрутизирующие пакетные коммутаторы с широкими функциональными возможностями без использования высокоинтегрированных коммуникационных микроконтроллеров.

Ключевые слова: пакетный коммутатор; интегрированный коммуникационный микроконтроллер; ИКМ; QUICC; PowerQUICC

1 Введение

В настоящее время у разработчиков телекоммуникационной аппаратуры, в частности устройств пакетной коммутации, большой популярностью пользуются интегрированные коммуникационные микроконтроллеры (ИКМ) [1]. Широко известные ИКМ семейств QUICC (QUad Integrated Communications Controller) и PowerQUICC впервые были разработаны компанией “Motorola”, а в настоящее время выпускаются ее преемницей на рынке микроэлектроники компанией “Freescale Semiconductor” [2–4]. Основная причина успеха ИКМ заключается в высокой степени интеграции как аппаратуры, так и функциональных возможностей на одном кристалле. Широкий набор этих возможностей уже обеспечил большое разнообразие областей применения ИКМ. Но в ряде приложений, особенно специального назначения, применение ИКМ может оказаться нежелательным по целому ряду причин, в том числе совершенно не технического характера. В этих ситуациях их приходится заменять менее интегрированными компонентами, т. е. как бы дезинтегрировать ИКМ, проигрывая при этом в объеме аппаратуры, а значит, надежности, потребляемой мощности и цене изделия. Следствием такой дезинтеграции может стать и снижение производительности разрабатываемого устройства.

При замене ИКМ менее интегрированными компонентами важно не просто уменьшить потери от дезинтеграции, но и добиться при этом каких-то ощутимых выигрышей, например, расширением функциональных возможностей устройства или повышением его пропускной способности. Один из возможных путей такой «компенсирующей» дезинтеграции был предложен в [5]. Суть его заключается в том, чтобы расширить «узкие места» архитектуры ИКМ, в частности распараллелить функции RISC-процессора коммуникационного модуля, распределив их между множеством простых микроконтроллеров, и разгрузить основную системную шину, разделив потоки инструкций программируемого ядра и коммутируемых данных.

Предлагаемая далее дезинтегрированная архитектура маршрутизирующего пакетного коммутатора практически реализует намеченные в [5] пути дезинтеграции ИКМ.

2 Основные компоненты дезинтегрированной архитектуры

Основные компоненты предлагаемой дезинтегрированной архитектуры универсального пакетного коммутатора и их взаимосвязь проиллюстрированы на рис. 1, где в качестве типичных примеров внешних портов коммутатора представлены следующие:

- низкоскоростной с интерфейсом E1;
- среднескоростной с интерфейсом Fast Ethernet (FE);
- высокоскоростной с интерфейсом Gigabit Ethernet (GE) .

В общем для предлагаемой архитектуры случае каждый внешний порт включает микропроцессор порта и два адаптера, которые могут быть реализованы на одной, как показано на рисунке, или разных программируемых логических интегральных

[1]Институт проблем информатики Российской академии наук, isokolov@ipiran.ru
[2]Институт проблем информатики Российской академии наук, vegorov@ipiran.ru

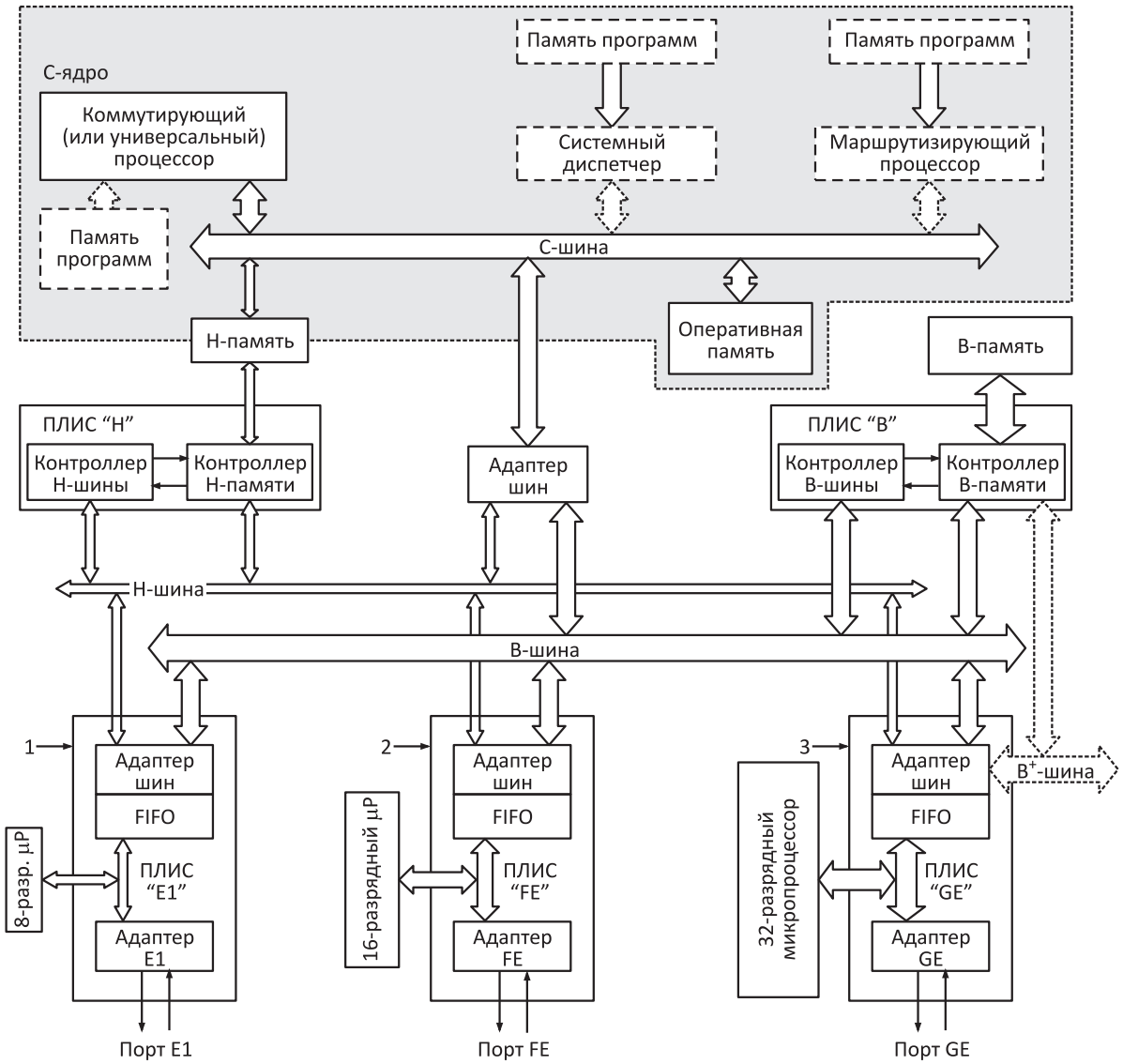


Рис. 1 Графическая иллюстрация дезинтегрированной архитектуры

схемах (ПЛИС). В последнем случае адаптер шин может быть унифицирован, и его варианты для различных портов и микропроцессоров будут отличаться лишь разрядностью внутренней шины данных.

Коммуникационные интерфейсы внешних портов коммутатора обслуживаются соответствующими адаптерами интерфейсов, которые обеспечивают необходимую аппаратную поддержку уровня MAC (Media Access Control) подключенного интерфейса. В частности, тракт передачи адаптера интерфейса должен оформлять и передавать, а тракт приема принимать, разграничивать и квалифицировать входящие блоки данных[1].

На рис. 1 не показаны компоненты физического уровня внешних интерфейсов, трудно реализуемые или совсем не реализуемые на типовых ПЛИС: приемопередатчики, импульсные трансформаторы, аналоговые усилители, пиковые детекторы, эквалайзеры и пр. Эти компоненты, обычно объединяемые в некие интегрированные трансиверы конкретных интерфейсов, являются внешними по отношению к предлагаемой архитектуре и несущественными для нее.

Каждый адаптер интерфейса обслуживается своим микропроцессором, производительность и реактивность которого должны быть достаточными, чтобы в совокупности с адаптером интерфей-

[1] В дальнейшем тексте для общности используется термин *блок данных*, но все сказанное в отношении них применимо к кадрам, ячейкам и т. п. конкретных MAC-интерфейсов.

са как минимум отработать протокол MAC-уровня подключенного интерфейса при принятых на том скоростях передачи данных.

Кроме того, существенной особенностью работы микропроцессора в предлагаемой дезинтегрированной архитектуре является необходимость отделения заголовков от тел входящих пакетов и добавление заголовков к телам исходящих пакетов. Отделяться и присоединяться могут заголовки или даже целые стеки заголовков любой длины, начиная от заголовков MAC-уровня и кончая целыми цепочками (стеками) заголовков, включающими заголовки сетевого, транспортного и даже сеансового уровней. Содержание отделяемых и присоединяемых заголовков и стеков заголовков не имеет значения для микропроцессора порта до тех пор, пока речь идет не об обработке, а о формальном их отделении и присоединении. Но в каких-то конкретных случаях микропроцессор порта может, не противореча описываемой архитектурной концепции, выполнять некую содержательную работу над отделяемыми заголовками, например фильтровать входящие блоки данных по MAC-адресам или меткам частных локальных сетей. Соответственно, требования к производительности микропроцессора порта оказываются весьма разными. Чтобы обеспечить требуемую производительность в зависимости от скорости передачи данных на порте и совокупной сложности выполняемых функций, микропроцессор порта может быть выбран 8-, 16- или 32-разрядным, с различными рабочими частотами, объемами памяти и другими особенностями.

Существенно, что предлагаемая дезинтегрированная архитектура сама по себе толерантна к типам и характеристикам микропроцессоров портов.

Если адаптеры интерфейсов и микропроцессоры портов могут варьироваться в самых широких пределах, то следующий компонент внешнего порта — адаптер шин — более универсален. Он обеспечивает сопряжение внешнего порта с двумя архитектурными шинами: «узкой» H-шиной заголовков (headers) и «широкой» B-шиной тел (bodies) блоков данных.

Принцип работы обеих шин одинаков, различаются они только шириной тракта данных и, соответственно, пропускной способностью. Поскольку по H-шине передаются относительно короткие заголовки, ее пропускная способность может быть меньше при небольшой ширине, в типичном случае — один байт. Ширина B-шины непосредственно определяет пропускную способность коммутатора, поэтому чем более высокая требуется пропускная способность, тем шире должна быть B-шина, в типичных случаях — от 32 до 128 разрядов.

Если у коммутатора имеется один или несколько сверхскоростных портов, например порт GE, то для адаптеров шин таких портов может потребоваться дополнительное локальное расширение B-шины (на рис. 1 показано пунктиром как B^+-шина). Архитектурные шины могут функционировать и на традиционном принципе временного слотового разделения (time-division mode) [6], и на принципе блочных передач, как, например, это было предложено в [7].

В более сложных случаях для обеспечения пропускной способности, превышающей возможности даже блочной шины, может быть использована та или иная коммутационная структура (switch fabric), что в целом не противоречит концепции предлагаемой архитектуры.

Входящие блоки данных разделяются аппаратурой тракта приема адаптера интерфейса и программным обеспечением микропроцессора порта на две части: H-блок, включающий заголовок или стек заголовков, и B-блок, содержащий остальную часть (тело) принятого блока данных. H-блок формируется немедленно после окончания приема отделяемого заголовка или стека заголовков и тут же отправляется по H-шине в память заголовков — H-память. B-блок формируется в процессе приема тела блока данных и по ходу приема отсылается по B-шине в память тел — B-память. В начало H- и B-блоков перед собственно данными вставляется начальный адрес буфера соответствующей памяти, в который должен быть помещен данный блок[1]. Возможно также явное указание действительной длины содержимого блока. Ссылка на буфер для тела блока данных в B-памяти также дублируется в H-блоке.

Очевидно, что B-память должна иметь объем и быстродействие, достаточные для складирования тел всех блоков данных, входящих со всех внешних портов, и их хранения до дальнейшей отправки. Например, если коммутатор имеет 20 портов и для каждого порта обеспечивается хранение до 100 пакетов длиной по 1,5 кбайт (типичная длина IP-пакета), то требуемый объем B-памяти составит 3 Мбайт. При ширине B-шины в 64 разряда и рабочей частоте 66 МГц, что типично для шин типа PCI (Peripheral Component Interconnect) и заведомо не превышает возможностей синхронной динамической памяти SDRAM (Synchronous Dynamic Random Access Memory), общая пропускная спо-

[1]Структура H- и B-блоков напоминает, например, структуру записываемого в память блока данных на шине PCI, где каждому такому блоку предшествует адрес этой памяти.

собность B-памяти, она же предельная пропускная способность всего коммутатора, равна приблизительно 2 Гбит/с. Если в коммутаторе требуются более высокие пропускные способности, например для обслуживания нескольких портов GE, то для такого случая предусматривается возможность локального расширения B^+-шиной. Пропускная способность архитектуры может быть повышена в несколько раз использованием B-памяти типа DDR (Double Data Rate) SDRAM и, сверх того, удвоена переходом к памяти QDR SRAM (Quad Data Rate Static Random Access Memory) с физическим разделением B-шины на два сепаратных тракта: тракта записи в память и тракта чтения из памяти. Наконец, при относительно небольшом числе высокоскоростных портов B-шина может быть заменена звездообразными дуплексными трактами, соединяющими контроллер B-памяти со всеми адаптерами шин.

Распределение ресурса B-шины между ее абонентами, т. е. портовыми адаптерами шины, выполняет контроллер B-шины, который может быть реализован отдельно или на одной ПЛИС совместно с контролером B-памяти (ПЛИС «B» в примере на рис. 1).

Отделяемые заголовки входящих пакетов должны содержать информацию, достаточную для принятия решения по коммутации соответствующего блока данных. В классической трактовке 7-уровневой модели ISO/OSI (International Standards Organization Open Systems Interconnection) для реализации простого коммутатора L2 достаточно отделять заголовки MAC-уровня. Включение в отделяемую часть заголовков сетевого и транспортного уровней предоставляет возможности более «интеллектуальной» коммутации L3 и L4.

Сразу после получения из тракта приема адаптера интерфейса заранее оговоренной необходимой для коммутации порции информации микропроцессор порта формирует H^I-блок, куда, помимо принятых данных, включает также адрес буфера в H-памяти для хранения этого H^I-блока и описатель буфера B-памяти, где будет храниться тело принимаемого «обезглавленного» блока данных. По мере формирования H^I-блоки отправляются по H-шине в двупортовую H-память, где они становятся доступными через ее второй порт коммутирующему процессору. При максимальном размере H^I-блока порядка 100 байт и принятых выше условиях по количеству внешних портов и числу складируемых блоков данных на порт минимальный требуемый объем H-памяти для хранения H^I-блоков равен приблизительно 1 Мбайт.

Аналогично B-шине, распределение ресурса H-шины между абонентами выполняет контроллер H-шины, который может быть реализован на одной ПЛИС (ПЛИС «H» на рис. 1) вместе с контролером H-памяти или совмещен с контроллером B-шины.

Коммутирующий процессор принимает решения по коммутации блоков данных на основании информации, получаемой в H^I-блоках, и формирует H^O-блоки, содержащие заголовки (стеки заголовков) для исходящих блоков данных и дескрипторы буферов с их телами в B-памяти. Эти H^O-блоки он возвращает обратно в H-память, откуда они по H-шине доставляются в микропроцессор порта назначения. Если полагать, что число исходящих из коммутатора пакетов примерно равно числу входящих, то и число проходящих через H-память H^O-блоков должно быть примерно равно числу хранящихся там же H^I-блоков. Соответственно, общий объем H-памяти при оговоренных выше условиях должен быть равен приблизительно 2 Мбайт.

Заметим, что поскольку тела складируемых блоков данных не попадают в оперативную память коммутирующего процессора, ее объем может быть относительно небольшим и целиком определяться нуждами программного обеспечения. Этот объем может оказаться совсем скромным, если коммутирующий процессор имеет гарвардскую архитектуру с отдельной памятью программ (на рис. 1 показана пунктиром).

Микропроцессор порта получает в H^O-блоке комплект заголовков исходящего блока данных и дескриптор буфера в B-памяти, где хранится тело этого блока. Это тело микропроцессор извлекает по B-шине, приклеивает к нему новые заголовки непосредственно из H^O-блока и передает вновь сформированный блок данных адаптеру интерфейса, тракт передачи которого оформляет блок надлежащим образом, сопровождая его преамбулой, флагами или символами SYNC, а также контрольной суммой и т. п.

Таким образом, предлагаемая дезинтегрированная архитектура расширяет сразу два «узких места» архитектуры ИКМ. Во-первых, вместо одного процессора коммуникационного модуля, обслуживающего все внешние порты ИКМ, она предполагает множество отдельных микропроцессоров, в результате чего на каждом порте может быть получена любая требуемая производительность по отработке канальных протоколов и, если требуется, протоколов более высокого уровня. При этом, в отличие от ИКМ, пропускная способность отдельного порта не зависит от загрузки других портов коммутатора. Во-вторых, в предлагаемой архитектуре предусмотрена отдельная память для хранения тел блоков данных и отдельная шина для их пересылки в эту память, вследствие чего два наиболее интенсивных потока информации в коммутаторе — поток ин-

струкций коммутирующего процессора и поток тел коммутируемых пакетов — никак не конфликтуют между собой.

Однако изоляция тел блоков данных в отдельном хранилище создает определенную проблему с административными блоками данных и блоками данных, инкапсулирующими пакеты коммутационных протоколов, содержимое которых должно быть доступно центральному ядру коммутатора (C-ядру на рис. 1). Это ядро в общем случае может включать, помимо коммутирующего процессора, системный диспетчер общего управления коммутатором и, в случае маршрутизирующего коммутатора, отдельный маршрутизирующий процессор. В частном случае функции какой-либо пары или всех трех перечисленных компонентов может выполнять один единственный универсальный процессор с достаточно высокой производительностью. Для решения проблемы доступа к B-памяти всех процессоров центрального ядра в предлагаемую архитектуру приходится вводить дополнительный адаптер шин, обеспечивающий сопряжение C- и B-шины. Если этот дополнительный адаптер реализован аналогично адаптерам шин внешних портов, то процессоры центрального ядра, помимо доступа к B-шине, «автоматически» получают доступ и к H-шине. Эта дополнительная возможность, как будет видно далее, оказывается полезной при инициализации коммутатора системным диспетчером, а также для тестовых и контрольно-диагностических целей.

3 Организация H- и B-шины

Особенность H- и B-шин состоит в том, что они должны обеспечить гарантированную полосу пропускания каждому порту коммутатора. При этом в зависимости от особенностей внешних интерфейсов и частных требований к коммутатору речь может идти как о среднем значении за какой-то период времени, так и о гарантии пропускания шиной очередного блока данных в очень жесткие интервалы времени, определяемые физической скоростью передачи данных на том или ином внешнем интерфейсе. Жесткость требований может быть в любой степени смягчена буферами FIFO (First In, First Out) соответствующего объема между адаптерами шин и микропроцессором порта, показанными на рис. 1, но принципиально это картины не меняет. С учетом сказанного для H- и B-шины можно предложить два решения:

(1) слотовая шина с жестким разделением временных слотов между портами коммутатора, как, например, это описано в [6];

(2) блочная шина, работающая по принципу сверхлокальной сети, в частности эффективное решение, предложенное в [7].

Слотовая шина, во-первых, способна обеспечить абонентам регулярный гарантированный доступ к памяти, а во-вторых, относительно проста и удобна в реализации. Однако, поскольку слотовая шина фактически реализует принцип коммутации каналов, ей свойственно присущее самому принципу недоиспользование потенциальной полосы пропускания. С этой точки зрения предпочтительнее выглядит блочная шина, реализующая, по существу, принцип пакетной коммутации; к тому же она лучше сопрягается с синхронной динамической памятью любого типа от простой SDRAM до QDR SRAM. Однако шина с блочной организацией заметно сложнее слотовой в реализации. Поэтому не стоит пренебрегать возможностью улучшить использование ресурса слотовой шины, варьируя в ней доли отдельных внешних портов в соответствии с максимальными скоростями передачи данных на них. Ниже кратко рассмотрен простой путь улучшения использования полосы пропускания простейшей слотовой шины.

Пусть слотовая B-шина, обслуживающая тех же трех абонентов, что показаны на рис. 1, имеет ширину 64 разряда[1], работает на частоте 50 МГц и разделена на 16 временных слотов[2]. При этих условиях пропускная способность одного временного слота составит 200 Мбит/с (25 Мбайт/с), а общая пропускная способность всей шины будет равна 3,2 Гбит/с (400 Мбайт/с). Три показанных на рис. 1 внешних порта поддерживают следующие максимальные (пиковые) дуплексные скорости передачи данных:

порт E1	2×2048 кбит/с (512 кбайт/с);
порт FE	2×100 Мбит/с (25 Мбайт/с);
порт GE	2×1 Гбит/с (250 Мбайт/с).

Тогда ресурс нашей гипотетической слотовой шины 3,2 Гбит/с можно поделить между внешними портами, например, следующим образом (см. рис. 2):

[1] В предположении, что все внешние порты, как и в нашем примере, дуплексные, с точки зрения дальнейших оценок не принципиально, рассматривается ли единая 64-разрядная шина или две сепаратные 32-разрядные шины, отдельно для чтения из B-памяти и записи в нее.

[2] Длительность временного слота шины желательно иметь кратной длине блока (burst length) SDRAM — в типичном минимальном варианте четырем тактам шины.

Рис. 2 Пример распределения ресурсов слотовой шины

№ слота	*i*								*i* + 1								*i* + 2		
Полуслот	Чтение				Запись				Чтение				Запись				Чтение		
Такт шины	R_1	R_2	R_3	R_4	W_1	W_2	W_3	W_4	R_1	R_2	R_3	R_4	W_1	W_2	W_3	W_4	R_1	R_2	
Передаваемая информация	Адрес H^O- или B-блока	(Адрес H^O- или B-блока)	(Адрес H^O- или B-блока)	(Адрес H^O- или B-блока)	Порция H^I- или B-блока	Порция H^I- или B-блока	Порция H^I- или B-блока	Порция H^I- или B-блока	Порция H^O- или B-блока	Порция H^O- или B-блока	Порция H^O- или B-блока	Порция H^O- или B-блока	Адрес H^I- или B-блока	(Адрес H^I- или B-блока)	(Адрес H^I- или B-блока)	(Адрес H^I- или B-блока)	|	|	
Индекс	*a*				*d*				*d*				*a*				—		
Занятость	●				●				●				●				○		

Рис. 3 Пример структуры слотов H- и B-шины

порт GE 12 слотов (300 кбайт/с);
порт FE 2 слота (50 Мбайт/с);
порт E1 1 слот (25 Мбайт/с).

Пример демонстрирует трудности с выделением временных слотов и относительную неэффективность слотовой шины. Для гарантии доступа и максимальной равномерности предоставления шины 12 слотов вместо минимально достаточных 10 выделено порту GE. По тем же причинам двойная полоса — два слота вместо минимально достаточного одного — выделена порту FE. Огромна и неизбежна при принятых условиях избыточность полосы для порта E1. В итоге на шине остался лишь один слот (слот 13), который может быть выделен для адаптера шины центрального ядра.

При небольшом числе внешних портов и ограниченных скоростях передачи на них отмеченная избыточность никому не мешает. Однако при большой потенциальной загрузке шины возникнет необходимость раздавать ее ресурсы более экономно. Уменьшать избыточность выделяемых полос можно простым уменьшением шага квантования, т. е. увеличением числа слотов в цикле шины.

Например, если бы цикл нашей шины состоял не из 16, а из 128 слотов, то минимальная выделяемая на порт полоса пропускания — квант полосы шины — уменьшилась бы с 25 до 3 Мбайт/с, что было бы достаточно для порта E1. Если в коммутаторе соседствуют низкоскоростные и высокоскоростные порты, то нужного эффекта можно достичь применением иерархических цикловых структур вроде суперциклов. Для нашего примера суперцикл из 32 циклов при 32 слотах в цикле уменьшил бы квант полосы шины до 40 кбайт/с.

Предлагаемая дезинтегрированная архитектура не накладывает жестких ограничений на структуру слотов H- и B-шины. Один из возможных примеров слотовой структуры со слотами размером в 8 тактов шины показан рис. 3.

В приведенном примере слот делится на два полуслота по 4 такта каждый. Первый полуслот (такты R_1–R_4) отводится на чтение данных адаптером шины соответствующего порта из H- или B-памяти, а второй полуслот (такты W_1–W_4) — на запись в H- или B-память. Таким образом, содержание передаваемой информации и направление ее передачи зависят от позиции полуслота. Кроме позиции определим еще два типа полуслотов: адресный — a-тип и информационный — d-тип с соответствующими индексами. Тип полуслота задается текущим мастером шины, в качестве которого выступает адаптер шины, контролирующий шину в данном слоте. В полуслоте чтения a-типа мастер

шины передает в контроллер памяти адрес буфера, из которого следует извлекать следующий H^O- или B-блок (слот i на рис. 3), а в полуслоте чтения d-типа контроллер памяти выдает мастеру шины очередную порцию H^O- или B-блока (слот $i+1$ на рис. 3). В полуслотах записи направление передачи всегда от мастера шины к контроллеру памяти: в полуслоте a-типа передается адрес буфера, в котором следует сохранять следующий H^I- или B-блок, а в полуслоте d-типа — очередная порция H^I- или B-блока.

На любой слотовой шине не каждый выделенный порту временной слот будет использоваться для передачи полезной информации. Пустые полуслоты неизбежны как из-за избыточности выделенных портам полос пропускания вследствие их квантования, так и просто из-за перерывов трафика на конкретном порте. Поэтому слотовая шина должна иметь отдельный маркер занятости полуслотов (на рис. 3 полуслот занят — «•», пуст — «○»). Маркер занятости устанавливается, раздельно для H- и B-шины, мастером шины, который тем самым получает возможность регулировать темп передачи данных независимо по обеим шинам, выдерживая при необходимости паузы как между H- и B-блоком, так и внутри них.

4 Организация буферов и адресация H- и B-памяти

Использовать абсолютные адреса на H- и B-шине неразумно. Дело не только в их разрядности. В целях взаимной защиты ресурсов, выделяемых для буферирования различным портам, желательно, чтобы каждый микропроцессор порта пользовался лишь относительными адресами внутри выделенных ему адресных пространств (пулов) H- и B-памяти. Выделение этих пространств должно быть прерогативой системного диспетчера C-ядра. Кроме того, начальные адреса буферов с разумными относительными потерями памяти могут задаваться на границах блоков размером 2^M байт, вследствие чего микропроцессор порта адресует буферы в категориях этих блоков, укорачивая тем самым на M разрядов адреса, передаваемые по H- и B-шине. Разумные значения M лежат в диапазоне 2–4 для H-памяти и, при типичной длине пакетов 1,5 кбайт, 4–8 для B-памяти.

В целом организация буферов в H- и B-памяти и их адресация со стороны микропроцессора могли бы быть следующими.

Системный диспетчер (или универсальный процессор) C-ядра в процессе инициализации коммутатора выделяет каждому внешнему порту в памяти некие пулы: по одному пулу в B-памяти — для B-блоков; и по два пула в H-памяти — для H^I- и H^O-блоков. Размер пула в B-памяти, выделяемого внешнему порту, должен быть, как правило, пропорционален общему объему проходящих через этот порт данных, т. е., в конечном счете, скорости передачи данных на порте. Размер пулов в H-памяти определяется выбранным размером пула в B-памяти с учетом типичного соотношения размеров H- и B-блоков для коммуникационных протоколов, принятых на данном порте.

Установочные параметры пулов системный диспетчер раздает микропроцессорам внешних портов через B-шину в форме неких S-блоков, по одному блоку на каждый внешний порт. S-блок включает базовые адреса и размеры всех выделенных для порта пулов, а также назначаемые порту временные слоты на H- и B-шине. Для передачи установочных параметров пулов и назначения слотов шины могут быть введены дополнительные типы (полу-) слотов, соответственно пуловые и слотовые. Как альтернатива для передачи установочных параметров могут использоваться адресные и информационные полуслоты, по умолчанию трактуемые иначе во время инициализации системы.

Мастером шины (шин) на все время инициализации во всех слотах является адаптер шин C-ядра. Поскольку передача S-блоков осуществляется по шине (шинам), слоты которой (которых) на данный момент еще не получили своего назначения, для инициализации системы должно использоваться некое исходное фиксированное (по умолчанию) соответствие слотов внешним портам. Простая и широко практикуемая основа установления такого фиксированного соответствия — «географическая» или любая другая физическая нумерация портов. В дальнейшем для определенности будет использоваться номер внешнего порта.

Пусть три внешних порта, показанных на рис. 1, имеют номера: порт E1 — 1, порт FE — 2, а порт GE — 3. На рис. 4 показан пример возможного использования H-шины во время инициализации коммутатора для назначения внешним портам слотов шины в соответствии с их распределением, показанным ранее на рис. 2.

Выдача системным диспетчером в слоте нуля означает, что этот слот либо не используется, либо резервируется для себя самим системным диспетчером. Ненулевой код в некотором слоте указывает номер внешнего порта, которому выделяется данный слот. Пример передачи портам установочных параметров пулов по B-шине приведен на рис. 5.

Первый такт каждого слота B-шины, такт P, несет номер внешнего порта, которому назнача-

Рис. 4 Пример использования H-шины для назначения слотов внешним портам

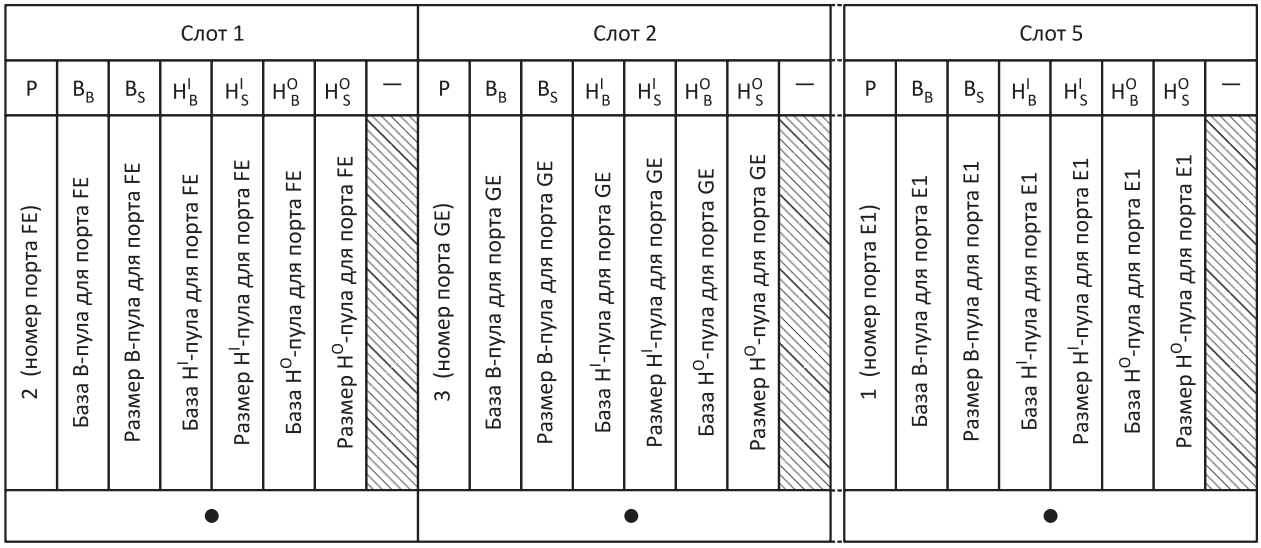


Рис. 5 Пример использования B-шины для передачи установочных параметров

ется данный слот, т. е. выполняет ту же роль, что и H-шина на рис. 4 (если, в частности, H-шина не задействована в процессе инициализации коммутатора). Информация в остальных тактах слота относится к внешнему порту, номер которого равен номеру данного слота, и включает базовые адреса и длины пулов, выделенные этому порту.

После рассылки установочных параметров коммутатор переходит в рабочий режим, при котором используется только относительная адресация пулов, организованных как циркулярные буферы. Для организации одного циркулярного буфера на одного абонента шины (т. е. на внешний порт или на системного диспетчера) контроллер соответствующей памяти должен иметь следующий комплект оборудования: регистр адресной базы, регистр размера пула, счетчик рабочих (относительных) адресов доступа, сумматор выходов счетчика с адресной базой пула и компаратор этих выходов с размером пула. Для обслуживания N абонентов шин ($N-1$ внешних портов плюс системный диспетчер) контроллер B-памяти должен иметь N таких комплектов оборудования, а контроллер H-памяти — $2N$ комплектов, хотя и несколько меньшей разрядности. Разумно однотипные элементы этих комплектов объединить внутри контроллера в блоки памяти с произвольным доступом объемом N слов каждый, унифицированно адресуемые номером текущего абонента. После объединения регистр адресной базы превращается в память адресных баз, регистр размера пула — в память размеров пулов, а счетчик относительных адресов доступа — в память относительных адресов. Общими остаются сумматор выходов счетчика с адресной базой пула и компаратор этих выходов с размером пула. Пример структуры контроллера B-памяти показан на рис. 6.

Память рабочих адресов загружается из полуслотов a-типа некими начальными адресами или просто нулями, которые в дальнейшем инкрементируются после каждого обращения к памяти, т. е. после каждого непустого полуслота d-типа данного абонента. Исполнительный адрес обращения к H- или B-памяти получается суммированием содержимого адресного счетчика, извлекаемого из памяти рабочих адресов, с базой пула, получаемой из памяти адресных баз. Для обеспечения циркулярности пулов контроллер каждой памяти должен в процессе инкрементации адресов сравнивать рабочий адрес с размером выделенного абоненту пула. Как только адрес после инкремента адресного счетчика переходит верхнюю границу пула, счетчик принудительно обнуляется, возвращая тем самым относительный адрес к началу пула.

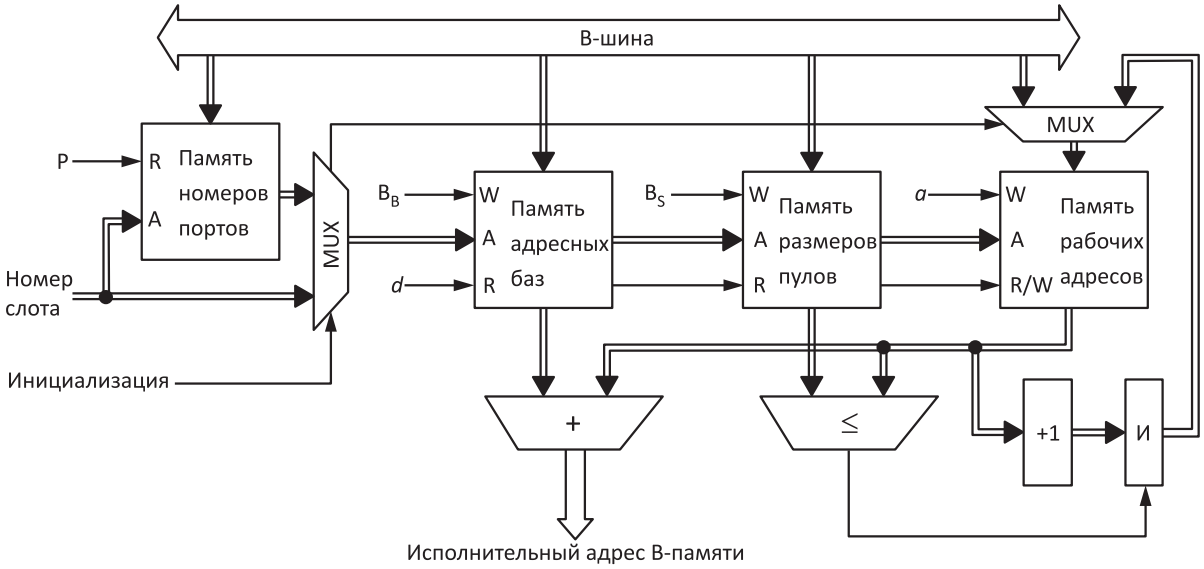


Рис. 6 Пример структуры контроллера B-памяти

В остальные памяти во время инициализации записываются соответственно номера внешних портов, адресные базы и длины пулов. Запись производится соответственно в тактах P, B_B и B_S B-шины (см. рис. 5). Адресом записи служит текущий номер слота B-шины. В рабочем режиме записанные в память назначения слотов номера внешних портов служат адресами для чтения адресных баз и размеров пулов, а также чтения и записи рабочих адресов. Поэтому в каждом слоте d-типа из всех памятей считывается информация, относящаяся к порту, которому был назначен данный слот. То же самое справедливо для стартовых адресов, заносимых в память рабочих адресов в полуслотах a-типа.

Практически удобно организовывать циркулярные пулы размером равным степеням двойки. В этом случае контроллеры шин могут обойтись без сумматоров для вычисления очередного адреса обращения и компараторов для контроля границ пулов. Адресная база пула значительно укорачивается и превращается в простой префикс для рабочего адреса, причем разрядность последнего устанавливается численно равной той самой назначенной для данного пула степени двойки. Префиксация заменяет суммирование адреса и базы, а переполнение счетчика автоматически возвращает текущий адрес на начало пула.

5 Заключение

Описанная дезинтегрированная архитектура пакетной коммутации реализует общую концепцию, предложенную в [5]. Она позволяет не только создавать пакетные коммутаторы без использования ИКМ, но и получать при этом ряд существенных преимуществ. В обсуждаемой дезинтегрированной архитектуре:

- снимаются присущие ИКМ серьезные и принципиальные ограничения на число внешних портов;
- устраняется зависимость пропускной способности отдельного внешнего порта от интенсивности трафиков на других портах;
- исключаются конфликты потоков инструкций и данных, присущие единой системной шине ИКМ;
- создаются предпосылки радикального повышения общей пропускной способности коммутатора и снижения времени коммутации пакетов.

Ограничение числа внешних портов в ИКМ является следствием нескольких причин.

Во-первых, всегда имеет место физическое ограничение числа выводов корпуса ИКМ. Дезинтеграция автоматически снимает эту проблему.

Во-вторых, в ИКМ все внешние порты обслуживаются одним коммуникационным процессором, реактивности которого недостаточно для обслуживания большого числа внешних портов [7]. По этой же причине в ИКМ неизбежна зависимость пропускной способности отдельного внешнего порта от интенсивности потоков коммутируемых данных на других портах. Несколько улучшает ситуацию применение для процессоров ИКМ специальных архитектур высокой реактивности [8, 9],

но лишь предлагаемая дезинтегрированная архитектура комплексно и радикально устраняет все отмеченные ограничения.

В-третьих, косвенно число обслуживаемых внешних портов зависит от общей пропускной способности коммутатора, а та, в свою очередь, от пропускной способности внутренних путей данных. Пространственно разделяя потоки инструкций коммутирующего процессора и коммутируемых им блоков данных, дезинтегрированная архитектура снимает и эту зависимость, открывая возможность найти более рациональное решение «по месту» для каждой специализированной шины или коммутационной структуры. В результате устраняются излишние задержки блоков данных внутри коммутатора и сокращается время коммутации.

Перечисленные преимущества обеспечивают дезинтегрированной архитектуре качественно новый уровень общей пропускной способности по сравнению с ИКМ. При этом уникальная пропускная способность с одинаковым успехом может быть реализована как в коммутаторах с небольшим числом высокоскоростных внешних портов, так и с большим числом низкоскоростных.

Предложенные в статье решения, безусловно, нельзя рассматривать как спецификацию новой архитектуры. В частности, не затронуты многие важные вопросы управления коммутатором в целом, а также ряд частных проблем коммутации и адаптации, таких как сегментация и сборка/разборка блоков данных, а также инкапсуляция пакетов, которые еще требуют своего решения. Но для поиска таких решений в рамках предлагаемой дезинтегрированной архитектуры на данный момент не видно никаких принципиальных препятствий.

Литература

1. *Шагурин И., Белецкий В.* Микроконтроллеры, интегрированные процессоры и гибридные DSP-процессоры компании FreeScale Semiconductors (SPS — Motorola) // Электронные компоненты, 2004. № 7.
2. *Егоров В. Б.* Принципы создания коммутационной аппаратуры на основе специализированных микроконтроллеров // Системы и средства информатики. — М.: Наука, 1999. Вып. 9. С. 44–55.
3. *Егоров В. Б.* Новое поколение коммуникационных микроконтроллеров компании “Freescale Semiconductor” // Chip News, 2007. No. 3. P. 14–18.
4. *Егоров В. Б.* Интегрированные коммуникационные процессоры компании “Freescale Semiconductor” // Электронные компоненты, 2007. № 8. С. 85–89.
5. *Соколов И. А., Егоров В. Б.* Дезинтеграционный подход к архитектуре универсального процессора коммутации пакетов // Информационные технологии и вычислительные системы, 2005. № 2. С. 76–85.
6. *Егоров В. Б.* Способ увеличения количества портов пакетного коммутатора с помощью слотовой шины // Информационные технологии и вычислительные системы, 2006. № 2. С. 16–21.
7. *Егоров В. Б., Полухин А. Н.* Принципы создания системной шины многопортовых пакетных коммутаторов // Системы и средства информатики. — М.: Наука, 2000. Вып. 10. С. 80–90.
8. *Егоров В. Б.* Способ повышения реактивности процессора // Информационные технологии и вычислительные системы, 2006. № 4. С. 3–15.
9. *Егоров В. Б.* «Многоэтажная» архитектура процессора // Информационные технологии и вычислительные системы, 2007. № 3. С. 79–87.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2008. Т. 2. Вып. 4. С. 12–47

МЕДИАННЫЕ МОДИФИКАЦИИ EM- И SEM-АЛГОРИТМОВ ДЛЯ РАЗДЕЛЕНИЯ СМЕСЕЙ ВЕРОЯТНОСТНЫХ РАСПРЕДЕЛЕНИЙ И ИХ ПРИМЕНЕНИЕ К ДЕКОМПОЗИЦИИ ВОЛАТИЛЬНОСТИ ФИНАНСОВЫХ ВРЕМЕННЫХ РЯДОВ*

А. К. Горшенин[1], В. Ю. Королёв[2], А. М. Турсунбаев[3]

Аннотация: Предложены медианные модификации EM- и SEM-алгоритмов и на примере численного решения задачи декомпозиции волатильности финансовых индексов демонстрируются их преимущества по сравнению с классическими методами. Приведены примеры декомпозиции волатильности различных финансовых временных рядов.

Ключевые слова: разделение смесей вероятностных распределений; робастность; эффективность; EM-алгоритм; SEM-алгоритм; волатильность

1 Введение

Для численного решения задачи разделения конечных смесей вероятностных распределений (т. е. задачи отыскания статистических оценок весов компонент смеси и параметров компонент смеси) при относительно большом числе компонент традиционно применяется EM-алгоритм. Если функция правдоподобия регулярна, то этот метод, как правило, находит наиболее правдоподобные оценки параметров смеси. Однако если функция правдоподобия нерегулярна, имеет много локальных максимумов (возможно, к тому же бесконечных), то EM-алгоритм становится крайне неустойчивым. К сожалению, последнее обстоятельство является серьезным препятствием при интерпретации результатов применения EM-алгоритма к разделению конечных смесей нормальных законов. Именно такие смеси повсеместно применяются при математическом моделировании многих явлений в самых разных областях — от биологии до экономики и от физики до финансового анализа.

В частности, как было экспериментально установлено, EM-алгоритм обладает сильной неустойчивостью по начальным данным. Например, в случае четырехкомпонентной смеси нормальных законов при объеме выборки 200–300 наблюдений замена лишь одного наблюдения другим может кардинально изменить итоговые оценки, полученные с помощью EM-алгоритма [1].

По-видимому, эта неустойчивость обусловлена тем, что стандартные (наиболее правдоподобные для случая нормального распределения) оценки математического ожидания и дисперсии (среднее арифметическое и выборочная дисперсия) при «засорении» (*контаминации*) выборки «посторонними» или «паразитными» наблюдениями становятся заметно менее эффективными по сравнению со, скажем, выборочной медианой. Этот эффект обнаружен Дж. Тьюки [2] и описан, например, в [3, 4]. Формально модель контаминации Тьюки сводится к тому, что вместо «чистого» модельного распределения, интерпретируемого как *однородная* модель, в качестве модельного распределения рассматривается неоднородная модель, имеющая вид смеси исходного «чистого» распределения и некоторого другого закона, описывающего «засоряющие» наблюдения. В задаче разделения смесей по сам́ой сути модели, когда оцениваются параметры одной компоненты смеси, наблюдения с распределениями, соответствующими другим компонентам, являются «загрязняющими». Это обстоятельство может сыграть особенно важную роль при реализации SEM-алгоритма, описываемого ниже. В данной статье в развитие методов, описанных в работе [5], предлагаются медианные модификации EM- и SEM-алгоритмов и на примере численного решения задачи декомпозиции волатильности финансовых индексов демонстрируются их преимущества по сравнению с классическими методами.

*Работа выполнена при поддержке РФФИ, гранты 08-01-00345, 08-01-00363, 08-07-00152.

[1]Московский государственный университет, факультет ВМиК, andygorshenin@gmail.com

[2]Московский государственный университет, факультет ВМиК, Институт проблем информатики РАН, vkorolev@comtv.ru

[3]Московский государственный университет, факультет ВМиК

2 EM-алгоритм для разделения смесей вероятностных распределений

Пусть $\boldsymbol{x} = (x_1, \ldots, x_n)$ — наблюдаемое значение случайной выборки $\boldsymbol{X} = (X_1, \ldots, X_n)$, в которой $X_1, \ldots, X_n$ — независимые случайные величины с одинаковой функцией распределения

$$F(x) = \sum_{i=1}^{k} p_i \Phi\left(\frac{x - a_i}{\sigma_i}\right), \quad x \in \mathbb{R}, \qquad (1)$$

где $a_i \in \mathbb{R}$, $\sigma_i > 0$, $p_i \geq 0$, $i = 1, \ldots, k$, $p_1 + \ldots + p_k = 1$, $\Phi(x)$ — стандартная нормальная функция распределения.

EM-алгоритмом принято называть итерационную процедуру поиска оценок максимального правдоподобия вектора θ параметров

$$\theta = (p_1, \ldots, p_k, a_1, \ldots, a_k, \sigma_1, \ldots, \sigma_k).$$

Применительно к смесям нормальных законов вида (1) EM-алгоритм определяется следующим образом (см., например, [1]). Пусть значение

$$\theta^{(m)} = \left(p_1^{(m)}, \ldots, p_k^{(m)}, a_1^{(m)}, \ldots, a_k^{(m)}, \sigma_1^{(m)}, \ldots, \sigma_k^{(m)}\right)$$

параметра θ на m-й итерации EM-алгоритма известно ($m \geq 0$). Обозначим

$$g_{ij}^{(m)} = \frac{\dfrac{p_i^{(m)}}{\sigma_i^{(m)}}\,\phi\left(\dfrac{x_j - a_i^{(m)}}{\sigma_i^{(m)}}\right)}{\displaystyle\sum_{r=1}^{k} \frac{p_r^{(m)}}{\sigma_r^{(m)}}\,\phi\left(\frac{x_j - a_r^{(m)}}{\sigma_r^{(m)}}\right)} = \frac{\dfrac{p_i^{(m)}}{\sigma_i^{(m)}}\exp\left\{-\dfrac{1}{2}\left(\dfrac{x_j - a_i^{(m)}}{\sigma_i^{(m)}}\right)^2\right\}}{\displaystyle\sum_{r=1}^{k} \frac{p_r^{(m)}}{\sigma_r^{(m)}}\exp\left\{-\frac{1}{2}\left(\frac{x_j - a_r^{(m)}}{\sigma_r^{(m)}}\right)^2\right\}}.$$

Величину $g_{ij}^{(m)}$ можно интерпретировать как статистическую оценку апостериорной вероятности того, что элемент X_j выборки сгенерирован в соответствии с i-й компонентой смеси (1) (т. е. $g_{ij}^{(m)}$ является «апостериорной вероятностью» того, что распределением случайной величины X_j является $\Phi\left(\left(x - a_i^{(m)}\right) \Big/ \sigma_i^{(m)}\right)$. Тогда значения параметров p_i, a_i и σ_i на $(m+1)$-й итерации EM-алгоритма соответственно определяются как

$$p_i^{(m+1)} = \frac{1}{n}\sum_{j=1}^{n} g_{ij}^{(m)};$$

$$a_i^{(m+1)} = \frac{1}{\displaystyle\sum_{j=1}^{n} g_{ij}^{(m)}} \sum_{j=1}^{n} g_{ij}^{(m)} x_j; \qquad (2)$$

$$\sigma_i^{(m+1)} = \left[\frac{1}{\displaystyle\sum_{j=1}^{n} g_{ij}^{(m)}} \sum_{j=1}^{n} g_{ij}^{(m)} \left(x_j - a_i^{(m+1)}\right)^2\right]^{1/2}, \quad i = 1, \ldots, k.$$

Обратим внимание на то, что $a_i^{(m+1)}$ является «выборочным средним», построенным по реализации $\boldsymbol{x} = (x_1, \ldots, x_n)$ выборки $\boldsymbol{X} = (X_1, \ldots, X_n)$, как если бы распределение каждого ее элемента задавалось вероятностями $g_{ij}^{(m)} / \sum_{j=1}^{n} g_{ij}^{(m)}$, $i = 1, \ldots, k$. EM-алгоритм довольно сильно зависит от начального приближения. Будучи алгоритмом проксимального типа [1, 6], он находит лишь локальный максимум функции правдоподобия. Для борьбы с этим недостатком предназначена, в частности, модификация EM-алгоритма, называемая стохастическим EM-алгоритмом или SEM-алгоритмом. Описание этого алгоритма будет специально приведено ниже.

EM-алгоритм также проявляет сильную неустойчивость по отношению к начальным данным. Для противодействия этому предназначены *медианные* модификации EM- и SEM-алгоритмов, которым, собственно, и посвящена данная статья. Строгое описание этих модификаций необходимо предварить обсуждением целесообразности применения в разных случаях разных — моментных и медианных — оценок параметров положения компонент смесей вида (1).

3 Относительная эффективность выборочного среднего и выборочной медианы при оценивании параметров положения компонент конечных смесей нормальных законов

Предположим, что в выборке $\boldsymbol{X} = (X_1, \ldots, X_n)$ все элементы независимы и имеют одну и ту же

непрерывную плотность распределения $f(x)$. Обозначим $m = \mathrm{med} X_1$. Предположим, что $f(m) > 0$. Пусть $\overline{m}_n$ — выборочная медиана, построенная по выборке $X_1, \ldots, X_n$. Еще в 1931 г. А. Н. Колмогоров [7] (см. также с. 111–114 в [8]) показал, что при $n \to \infty$

$$\mathsf{P}\big(\sqrt{n}(\overline{m}_n - m) < x\big) \longrightarrow \Phi\big(2f(m)x\big)\,,$$

где, как обычно, $\Phi(y)$ — стандартная нормальная функция распределения, так что $\Phi\big(2f(m)x\big)$ — функция распределения нормально распределенной случайной величины с дисперсией $\big(2f(m)\big)^{-2}$. Таким образом, для рассмотренного выше критерия качества выборочной медианы при больших n имеем

$$\mathsf{E}(\overline{m}_n - m)^2 = \frac{1}{n}\mathsf{E}\left[\sqrt{n}(\overline{m}_n - m)\right]^2 \approx \frac{1}{4n\,(f(m))^2}\,.$$

Если дополнительно обозначить $a = \mathsf{E}X_1$, $\overline{X} = (1/n)\sum\limits_{j=1}^{n} X_j$, то согласно центральной предельной теореме

$$\mathsf{P}\left(\sqrt{n}\,(\overline{X}_n - a) < x\right) \longrightarrow \Phi\left(\frac{x}{\sqrt{\mathsf{D}X_1}}\right)\,,$$

т. е. при больших n

$$\mathsf{E}(\overline{X}_n - a)^2 = \frac{1}{n}\mathsf{E}\left[\sqrt{n}(\overline{X}_n - a)\right]^2 \approx \frac{\mathsf{D}X_1}{n}\,.$$

Таким образом, ответ на вопрос о том, какая из оценок — выборочное среднее или выборочная медиана — лучше, можно получить, скажем, вычислив отношение

$$\frac{\mathsf{E}(\overline{X}_n - a)^2}{\mathsf{E}(\overline{m}_n - m)^2} \approx 4\,(f(m))^2\,\mathsf{D}X_1$$

(относительную эффективность оценок $\overline{X}_n$ и $\overline{m}_n$).

В частности, если $f(x)$ — плотность нормального распределения со средним a и дисперсией $\mathsf{D}X_1$:

$$f(x) = \frac{1}{\sqrt{2\pi\mathsf{D}X_1}}\exp\left\{-\frac{(x-a)^2}{2\mathsf{D}X_1}\right\}\,,$$

то, во-первых, $a = m$ и, во-вторых, $f(m) = 1/\sqrt{2\pi\mathsf{D}X_1}$, так что

$$\frac{\mathsf{E}(\overline{X}_n - a)^2}{\mathsf{E}(\overline{m}_n - m)^2} \approx \frac{2}{\pi}\,.$$

Очевидно, что если в нормальном случае для оценивания параметра положения использовать выборочную медиану, то для того, чтобы достичь той же точности, что при использования выборочного среднего, понадобится в $\pi/2 \approx 1.57$ раз больше наблюдений, т. е. в таком случае выборочная медиана примерно в полтора раза менее эффективна, нежели выборочное среднее.

При использовании выборочного среднего и выборочной медианы в качестве статистических оценок параметра, характеризующего «центр» распределения, следует заметить, что выборочная медиана обладает большей устойчивостью к присутствию в выборке так называемых «загрязняющих» наблюдений. Действительно, если выборка $X_1, \ldots, X_n$ в некотором смысле не является однородной, т. е. наряду с наблюдениями, имеющими функцию распределения $F(x)$, в ней присутствуют наблюдения с какой-то другой функцией распределения, то в выборочное среднее наряду с «правильными» наблюдениями войдут *значения* «загрязняющих» наблюдений. При этом если значения «загрязняющих» наблюдений велики, то их присутствие, естественно, сильно смажет итоговую картину. В то же время отклонения выборочной медианы от ее «правильного» значения зависят не столько от значений «загрязняющих» наблюдений, сколько от их числа. Такое свойство выборочной медианы, как известно, называется робастностью.

Вышеупомянутое свойство робастности выборочной медианы хорошо иллюстрируется на примере следующей ситуации. Предположим, что в независимой выборке $X_1, \ldots, X_n$ все элементы имеют одну и ту же плотность распределения

$$f(x) = \sum_{i=1}^{k} \frac{p_i}{\sqrt{2\pi}\cdot\sigma_i}\exp\left\{-\frac{(x-a)^2}{2\sigma_i^2}\right\}\,,$$

где $0 < p_i < 1$, $i = \overline{1,k}$, $p_1 + \ldots + p_k = 1$ и $\sigma_i^2 > 0$. Эту ситуацию можно интерпретировать как наличие в выборке примерно $p_i \cdot 100\%$ наблюдений с нормальным распределением, имеющим параметры a и σ_i^2, $i = \overline{1,k}$, т. е. изучаемая популяция (генеральная совокупность) является *смесью* k популяций, каждая из которых нормально распределена с параметрами a и σ_i^2, причем доли этих k субпопуляций (*компонент смеси*) составляют соответственно $p_i \cdot 100\%$, $i = \overline{1,k}$. Если при этом какое-либо из значений p_i близко к единице, то говорят, что выборка из i-й субпопуляции *загрязнена* объектами (наблюдениями) из других субпопуляций. Заметим, что параметры «центра» у всех компонент смеси одинаковы. Легко видеть, что $a = m$ и

$$f(m) = \frac{1}{\sqrt{2\pi}}\sum_{i=1}^{k}\frac{p_i}{\sigma_i}.$$

Далее,

$$\mathsf{D}X_1 = \int\limits_{-\infty}^{\infty} (x-a)^2 \sum_{i=1}^{k} \frac{p_i}{\sqrt{2\pi}\cdot\sigma_i} \exp\left\{-\frac{(x-a)^2}{2\sigma_i^2}\right\} dx = \sum_{i=1}^{k} \frac{p_i}{\sqrt{2\pi}\sigma_i} \int\limits_{-\infty}^{\infty} (x-a)^2 \exp\left\{-\frac{(x-a)^2}{2\sigma_i^2}\right\} dx = \sum_{i=1}^{k} p_i\sigma_i^2 \,.$$

Вычислим асимптотическую относительную эффективность выборочного среднего и выборочной медианы:

$$\lim_{n\to\infty} \frac{\mathsf{E}\left(\overline{X}_n - a\right)^2}{\mathsf{E}\left(\overline{m}_n - m\right)^2} = 4\left(f(m)\right)^2 \mathsf{D}X_1 = \frac{2}{\pi}\left(\sum_{i=1}^{k}\frac{p_i}{\sigma_i}\right)^2 \sum_{i=1}^{k} p_i\sigma_i^2 \,.$$

Несложно видеть, что если зафиксировать все параметры $p_1,\ldots,p_k,\sigma_1,\ldots,\sigma_k$, кроме одного, скажем σ_{i_0}, то правая часть последнего соотношения неограниченно возрастает при неограниченном увеличении σ_{i_0}. Действительно,

$$\frac{\pi}{2}\lim_{n\to\infty}\frac{\mathsf{E}(\overline{X}_n - a)^2}{\mathsf{E}(\overline{m}_n - m)^2} = \left(\frac{p_{i_0}}{\sigma_{i_0}} + A\right)^2 \left(p_{i_0}\sigma_{i_0}^2 + B\right) = A^2 p_{i_0}\sigma_{i_0}^2 + 2Ap_{i_0}^2\sigma_{i_0} + 2AB\frac{p_{i_0}}{\sigma_{i_0}} + B\frac{p_{i_0}^2}{\sigma_{i_0}^2} + p_{i_0}^3 + A^2B\,, \quad (3)$$

где

$$A = \sum_{\substack{1\le i\le k\\ i\ne i_0}} \frac{p_i}{\sigma_i}\,; \quad B = \sum_{\substack{1\le i\le k\\ i\ne i_0}} p_i\sigma_i^2$$

и первые два слагаемых в правой части (3) неограниченно возрастают при $\sigma_{i_0}\to\infty$, в то время как остальные слагаемые стремятся к $p_{i_0}^3 + A^2B$.

К примеру, если $k=2$, $\sigma_1=1$, $p_1=0{,}01$, $p_2=0{,}99$, то выборочная медиана эффективнее выборочного среднего для $\sigma_2^2 > 61$. Если же $p_1=0{,}05$, то выборочная медиана эффективнее выборочного среднего для $\sigma_2^2>14$. Наконец, если доля «загрязняющих» наблюдений составляет 10%, то выборочная медиана эффективнее выборочного среднего уже для $\sigma_2^2>9{,}1$.

4 Медианные модификации EM-алгоритма

Как было экспериментально установлено, EM-алгоритм обладает сильной неустойчивостью по начальным данным. Например, в случае четырехкомпонентной смеси нормальных законов при объеме выборки 200–300 наблюдений замена лишь одного наблюдения другим может кардинально изменить итоговые оценки, полученные с помощью EM-алгоритма.

Для противодействия указанной неустойчивости EM-алгоритма можно использовать его медианные модификации. Смысл этих модификаций в том, что наиболее «неустойчивые» этапы выполнения EM-алгоритма заменяются более устойчивыми. В частности, на M-этапе неустойчивые моментные оценки наибольшего правдоподобия (которые для нормальных компонент минимизируют квадратичный риск) заменяются более устойчивыми (робастными) оценками медианного типа, оптимальными в смысле среднего абсолютного отклонения. Более того, в задачах разделения смесей вероятностных распределений выборочные медианы при определенных соотношениях между параметрами смеси оказываются более эффективными, нежели оценки максимального правдоподобия типа выборочных моментов, т. е. иногда выборочные медианы оптимальны не только в смысле среднего абсолютного отклонения, но и в традиционном смысле квадратичного риска.

Опишем две возможные медианные модификации M-этапа EM-алгоритма. В рамках этих модификаций параметры a_i оцениваются одинаково. Различными являются лишь оценки параметров σ_i.

Пусть числа $g_{ij}^{(m)}$ известны. По числам $g_{ij}^{(m)}$ определим «вероятности» $p_{ij}^{(m)}$ по правилу

$$p_{ij}^{(m)} = g_{ij}^{(m)}\left(\sum_{j=1}^{n} g_{ij}^{(m)}\right)^{-1}, \quad i=1,\ldots,k;\ j=1,\ldots,n$$

(n — объем выборки, k — число компонент смеси). Пусть $\boldsymbol{x}=(x_1,\ldots,x_n)$ — выборка. Тогда число $p_{ij}^{(m)}$ можно интерпретировать как вероятность того, что наблюдение x_j имеет распределение, определяемое i-й компонентой смеси.

Введем «фиктивные» случайные величины $\xi_i^{(m)}$, $i=1,\ldots,k$, которые соответственно принимают значение x_j с вероятностями $p_{ij}^{(m)}$, $i=1,\ldots,k$, $j=1,\ldots,n$ (несложно видеть, что $\sum\limits_{j=1}^{n} p_{ij}^{(m)} = 1$). При этом оценка параметра сдвига i-й компоненты смеси на $(m+1)$-й итерации, приведенная в

предыдущем разделе, оказывается в точности равной математическому ожиданию случайной величины $\xi_i^{(m)}$:

$$a_i^{(m+1)} = \frac{1}{\sum\limits_{j=1}^{n} g_{ij}^{(m)}} \sum_{j=1}^{n} g_{ij}^{(m)} x_j = \sum_{j=1}^{n} p_{ij}^{(m)} x_j = \mathsf{E}_{\theta^{(m)}} \xi_i^{(m)} .$$

Для того чтобы построить модификацию EM-алгоритма, более устойчивую по отношению к наличию «засоряющих» наблюдений (а при оценивании параметров какой-либо компоненты смеси наблюдения, распределения которых соответствуют другим компонентам, неизбежно будут «засоряющими» по отношению к оцениваемой компоненте), в качестве оценки параметра a_i на $(m+1)$-й итерации предлагается взять медиану $\operatorname{med} \xi_i^{(m)}$ случайной величины $\xi_i^{(m)}$, которую можно вычислить так. Переупорядочим значения $x_1, \ldots, x_n$ случайной величины $\xi_i^{(m)}$ по неубыванию. Получим вариационный ряд $x_{(1)}, \ldots, x_{(n)}$. Ясно, что одно и то же переупорядочение имеет место для значений всех случайных величин $\xi_i^{(m)}$. Одновременно переставятся и вероятности $p_{ij}^{(m)}$, соответствующие значениям каждой случайной величины $\xi_i^{(m)}$. Пусть $\widehat{p}_{ij}^{(m)}$ — это та из вероятностей $p_{ij}^{(m)}$, которая соответствует значению $x_{(j)}$ случайной величины $\xi_i^{(m)}$. Положим

$$J_i = \min \left\{ j : \widehat{p}_{i1}^{(m)} + \widehat{p}_{i2}^{(m)} + \ldots + \widehat{p}_{ij}^{(m)} \geq \frac{1}{2} \right\} .$$

Тогда

$$a_i^{(m+1)} = \operatorname{med} \xi_i^{(m)} = x_{(J_i)} . \qquad (4)$$

Для оценивания параметра σ_i на $(m+1)$-й итерации сначала по указанной выше схеме вычислим медиану случайной величины $\left| \xi_i^{(m)} - a_i^{(m+1)} \right|$,

$$\widehat{m}_i^{(m+1)} = \operatorname{med} \left| \xi_i^{(m)} - a_i^{(m+1)} \right| .$$

Затем введем «фиктивную» случайную величину $\zeta_i^{(m+1)}$ с функцией распределения

$$\mathsf{P}_{\theta^{(m+1)}} \left(\zeta_i^{(m+1)} < x \right) = \Phi \left(\frac{x - a_i^{(m+1)}}{\sigma_i^{(m+1)}} \right) ,$$

т. е. распределение случайной величины $\zeta_i^{(m+1)}$ является i-й компонентой смеси. «Эмпирическим» аналогом случайной величины $\zeta_i^{(m+1)}$ является случайная величина $\xi_i^{(m)}$, введенная ранее. В идеале (при достаточно большом m и при большом n) должно быть справедливо приближенное равенство $\mathsf{P}_{\theta^{(m+1)}} \left(\zeta_i^{(m+1)} < x \right) \approx \mathsf{P}_{\theta^{(m+1)}} \left(\xi_i^{(m)} < x \right)$, $-\infty < x < +\infty$.

Таким образом, отыскав эмпирическую медиану $\widehat{m}_i^{(m+1)}$ (т. е. медиану случайной величины $\left| \xi_i^{(m)} - a_i^{(m+1)} \right|$), в соответствии с идеологией метода моментов можно сказать, что она близка к медиане $\mu_i^{(m+1)}$ случайной величины $\left| \zeta_i^{(m+1)} - a_i^{(m+1)} \right|$.

Медиана $\mu_i^{(m+1)}$ случайной величины $\left| \zeta_i^{(m+1)} - a_i^{(m+1)} \right|$ определяется из условия

$$\mathsf{P}_{\theta^{(m+1)}} \left(\left| \zeta_i^{(m+1)} - a_i^{(m+1)} \right| \leq \mu_i^{(m+1)} \right) = \frac{1}{2} .$$

Но

$$\begin{aligned} \mathsf{P}_{\theta^{(m+1)}} \left(\left| \zeta_i^{(m+1)} - a_i^{(m+1)} \right| \leq \mu_i^{(m+1)} \right) &= \\ = \mathsf{P}_{\theta^{(m+1)}} \left(-\mu_i^{(m+1)} \leq \zeta_i^{(m+1)} - a_i^{(m+1)} \leq \right. & \\ \left. \leq \mu_i^{(m+1)} \right) = \mathsf{P}_{\theta^{(m+1)}} \left(a_i^{(m+1)} - \mu_i^{(m+1)} \leq \right. & \\ \left. \leq \zeta_i^{(m+1)} \leq a_i^{(m+1)} + \mu_i^{(m+1)} \right) = & \\ = \Phi \left(\frac{\left(a_i^{(m+1)} + \mu_i^{(m+1)} \right) - a_i^{(m+1)}}{\sigma_i^{(m+1)}} \right) - & \\ - \Phi \left(\frac{\left(a_i^{(m+1)} - \mu_i^{(m+1)} \right) - a_i^{(m+1)}}{\sigma_i^{(m+1)}} \right) = & \\ = \Phi \left(\frac{\mu_i^{(m+1)}}{\sigma_i^{(m+1)}} \right) - \Phi \left(-\frac{\mu_i^{(m+1)}}{\sigma_i^{(m+1)}} \right) = & \\ = 2\Phi \left(\frac{\mu_i^{(m+1)}}{\sigma_i^{(m+1)}} \right) - 1 . & \end{aligned}$$

Следовательно, справедливо соотношение

$$2\Phi \left(\frac{\mu_i^{(m+1)}}{\sigma_i^{(m+1)}} \right) - 1 = \frac{1}{2} ,$$

т. е.

$$\Phi \left(\frac{\mu_i^{(m+1)}}{\sigma_i^{(m+1)}} \right) = \frac{3}{4} ,$$

что эквивалентно соотношению

$$\frac{\mu_i^{(m+1)}}{\sigma_i^{(m+1)}} = u_{3/4} ,$$

где $u_{3/4}$ — квантиль порядка $3/4$ стандартного нормального закона. В таблицах находим $u_{3/4} \approx 0{,}6745$. Следуя идеологии метода моментов, приравняем эмпирическую медиану $\widehat{m}_i^{(m+1)}$ теоретической медиане $\mu_i^{(m+1)}$ и окончательно получим уравнение для оценки параметра σ_i на $(m+1)$-й итерации:

$$\sigma_i^{(m+1)} = \frac{\widehat{m}_i^{(m+1)}}{u_{3/4}} = 1{,}4826\,\widehat{m}_i^{(m+1)}. \qquad (5)$$

Оценки $p_i^{(m+1)}$ весов p_i в модели (1) ищутся, как и ранее, по формулам (2). Числа же $g_{ij}^{(m+1)}$ на каждой итерации переназначаются так же, как и ранее, а именно

$$g_{ij}^{(m+1)} = \frac{\dfrac{p_i^{(m+1)}}{\sigma_i^{(m+1)}} \exp\left\{ -\frac{1}{2}\left(\frac{x_j - a_i^{(m+1)}}{\sigma_i^{(m+1)}} \right)^2 \right\}}{\displaystyle\sum_{r=1}^{k} \frac{p_r^{(m+1)}}{\sigma_r^{(m+1)}} \exp\left\{ -\frac{1}{2}\left(\frac{x_j - a_r^{(m+1)}}{\sigma_r^{(m+1)}} \right)^2 \right\}}. \qquad (6)$$

Итак, соотношения (2), (4)–(6) определяют первую медианную модификацию EM-алгоритма.

Вторая медианная модификация EM-алгоритма отличается от первой лишь способом оценивания параметров σ_i. А именно: вычислим $\mathsf{E}_{\theta^{(m+1)}} \left| \zeta_i^{(m+1)} - a_i^{(m+1)} \right|$. Имеем

$$\mathsf{E}_{\theta^{(m+1)}} \left| \zeta_i^{(m+1)} - a_i^{(m+1)} \right| = \int_{-\infty}^{\infty} \left| x - a_i^{(m+1)} \right| d_x \Phi\left(\frac{x - a_i^{(m+1)}}{\sigma_i^{(m+1)}} \right) = 2 \int_0^{\infty} x\, d_x \Phi\left(\frac{x}{\sigma_i^{(m+1)}} \right) = \sigma_i^{(m+1)} \sqrt{\frac{2}{\pi}}.$$

Эмпирическим аналогом величины $\mathsf{E}_{\theta^{(m+1)}} \left| \zeta_i^{(m+1)} - a_i^{(m+1)} \right|$ является величина

$$s_i^{(m+1)} = \mathsf{E}_{\theta^{(m)}} \left| \xi_i^{(m)} - a_i^{(m+1)} \right| = \sum_{j=1}^{n} p_{ij}^{(m)} \left| x_j - a_i^{(m+1)} \right|.$$

Реализуя метод моментов и приравнивая величину $\mathsf{E}_{\theta^{(m+1)}} \left| \zeta_i^{(m+1)} - a_i^{(m+1)} \right|$ ее эмпирическому аналогу, получаем еще одну оценку для параметра σ_i на $(m+1)$-й итерации:

$$\sigma_i^{(m+1)} = \sqrt{\frac{\pi}{2}} \cdot s_i^{(m+1)} = 1{,}2533 s_i^{(m+1)}. \qquad (7)$$

Таким образом, вторая медианная модификация EM-алгоритма определяется соотношениями (2), (4), (7) и (6).

Заметим, что вторая модификация более соответствует духу так называемой L_1-теории устойчивого оценивания в силу известного свойства

$$\arg\min_a \mathsf{E}_{\theta^{(m+1)}} \left| \zeta_i^{(m+1)} - a \right| = \operatorname{med} \zeta_i^{(m+1)} \left(\approx \operatorname{med} \xi_i^{(m)} = a_i^{(m+1)} \right).$$

5 SEM-алгоритм

Классический EM-алгоритм выбирает первый попавшийся локальный максимум, т. е., являясь методом локальной оптимизации, он приводит не к глобальному максимуму функции правдоподобия, а к тому локальному максимуму, который является ближайшим к начальному приближению.

Самый простой способ противодействия этому свойству заключается в том, чтобы, не ограничиваясь единственным начальным приближением и, соответственно, единственной траекторией EM-алгоритма, реализовать несколько траекторий, задавая (например, случайно) несколько различных начальных приближений, а затем выбрать тот из результатов, для которого правдоподобие является наибольшим среди всех реализованных траекторий EM-алгоритма. Однако при таком подходе остается неясным ответ на вопрос о том, каким механизмом разумнее всего пользоваться при переходе от одного начального приближения к другому. В частности, когда начальное приближение задается случайно, без дополнительной информации нельзя исчерпывающим образом определить распределение вероятностей, в соответствии с которым следует генерировать очередное начальное приближение.

Другой, оказавшийся весьма эффективным, способ заключается как бы в случайном «встряхивании» наблюдений (выборки) на каждой итерации. Этот способ лежит в основе SEM-алгоритма, название которого является аббревиатурой термина *Stochastic EM-algorithm* (стохастический (или случайный) EM-алгоритм) [1].

Чтобы описать SEM-алгоритм, представим ненаблюдаемую информацию в иной форме (однако, по сути, эквивалентной старой форме). А именно: будем считать, что каждому наблюдению x_j соответствует вектор $\vec{y}_j = (y_{1j}, y_{2j}, \ldots, y_{kj})$, $j = 1, \ldots, n$,

где k — число компонент смеси, n — объем выборки. При этом

$$y_{ij} = \begin{cases} 1\,, & \text{если наблюдение } x_j \\ & \text{порождено } i\text{-й компонентой смеси}\,; \\ 0\,, & \text{в противном случае.} \end{cases}$$

При каждом j только одна из компонент вектора $\vec{y}_j$ равна единице, остальные компоненты этого вектора равны нулю.

В терминах величин $\mathbf{y} = \{\vec{y}_j = (y_{1j}, y_{2j}, \ldots, y_{kj}), j = 1, \ldots, n\}$ логарифм полной функции правдоподобия для модели (1) принимает вид

$$\log L(\theta; x, y) = \sum_{j=1}^{n}\sum_{i=1}^{k} y_{ij} \log\left[p_i \psi_i(x_j; t_i)\right] =$$
$$= \sum_{i=1}^{k} \log p_i \sum_{j=1}^{n} y_{ij} + \sum_{i=1}^{k}\sum_{j=1}^{n} y_{ij} \log \psi_i(x_j; t_i)\,. \quad (8)$$

Векторы $\vec{y}_j = (y_{1j}, y_{2j}, \ldots, y_{kj})$, $j = 1, \ldots, n$, разбивают исходную наблюдаемую выборку $\mathbf{x}$ на k классов (кластеров) $K_1, \ldots, K_k$:

$$\mathbf{x} = K_1 \cup \ldots \cup K_k\,.$$

Для каждого $i = 1, \ldots, k$ с формальной точки зрения K_i — это множество тех наблюдений x_j, каждому из которых соответствует $y_{ij} = 1$. При этом каждое наблюдение x_j входит ровно в один кластер, т. е. $K_i \cap K_j = \varnothing$ при $i \neq j$. Пусть υ_i — это число наблюдений, попавших в кластер K_i, $i = 1, \ldots, k$,

$$\upsilon_i = \sum_{j=1}^{n} y_{ij}\,.$$

Очевидно, что $\upsilon_1 + \ldots + \upsilon_k = n$. Тогда, продолжая (8), для логарифма полной функции правдоподобия в модели (1) получаем представление

$$\log L(\theta; x, y) = \sum_{i=1}^{k} \upsilon_i \log p_i + \sum_{i=1}^{k} \sum_{j:\, x_j \in K_i} \log \psi_i(x_j; t_i)\,. \quad (9)$$

Если бы величины y_{ij} были известны, то искать значение θ, максимизирующее функцию правдоподобия (9), можно было бы, максимизируя по θ каждое из слагаемых в правой части (9), поскольку эти слагаемые зависят только от «своих» групп параметров. А именно: с помощью метода неопределенных множителей Лагранжа несложно убедиться, что максимум первого слагаемого по набору $p_1, \ldots, p_k$ при очевидном ограничении $p_1 + \ldots + p_k = 1$ достигается при

$$p_i^* = \frac{\upsilon_i}{n}\,. \quad (10)$$

Далее заметим, что

$$\sum_{j:\, x_j \in K_i} \log \psi_i(x_j; t_i) = \log \prod_{j:\, x_j \in K_i} \psi_i(x_j; t_i) \equiv \log L_i(t_i; K_i),$$

где $L_i(t_i; K_i)$ — это функция правдоподобия параметра t_i, построенная по подвыборке (кластеру) K_i в предположении, что каждый элемент подвыборки имеет плотность распределения $\psi_i(x_j; t_i)$. Отсюда видно, что значения

$$t_i^* = \arg\max L_i(t_i; K_i),\ i = 1, \ldots, k\,, \quad (11)$$

доставляют максимум второму слагаемому в правой части (9). Легко видеть, что соотношение (11) определяет обычные оценки наибольшего правдоподобия для параметров i-й компоненты смеси (1), построенные по подвыборке наблюдений, распределение которых равно этой компоненте, т. е. по кластеру K_i.

Таким образом, если бы величины y_{ij} были известны, то оценки наибольшего правдоподобия параметров модели (1) определялись бы соотношениями (10) и (11). Однако на практике величины y_{ij} неизвестны. Идея SEM-алгоритма заключается в том, что эти величины определяются с помощью специального имитационного моделирования.

Итерационный SEM-алгоритм определяется так. Предположим, что известны значения $g_{ij}^{(m)}$ апостериорных вероятностей принадлежности наблюдения x_j к кластеру K_i, $i = 1, \ldots, k$; $j = 1, \ldots, n$; m — номер итерации (отметим, что $\sum\limits_{i=1}^{k} g_{ij}^{(m)} = 1$ для каждого j и при каждом m).

На первом этапе SEM-алгоритма (*S-этапе*, от слов *Stochastic* или *Simulation*) для каждого $j = 1, \ldots, n$ генерируются векторы $\vec{y}_j^{\,(m+1)} = \left(y_{1j}^{(m+1)}, y_{2j}^{(m+1)}, \ldots, y_{kj}^{(m+1)}\right)$ как реализации случайных векторов с полиномиальным распределением с параметрами 1 и $g_{1j}^{(m)}, \ldots, g_{kj}^{(m)}$ ($g_{ij}^{(m)}$ — это вероятность того, что $y_{ij}^{(m+1)} = 1$). По векторам $\vec{y}_j^{\,(m+1)}$ определяется разбиение выборки $\mathbf{x} = (x_1, \ldots, x_n)$ на кластеры $K_1^{(m+1)}, \ldots, K_k^{(m+1)}$ и соответствующие числа $\upsilon_1^{(m+1)}, \ldots, \upsilon_k^{(m+1)}$ (численности кластеров) на $(m+1)$-й итерации. (Можно сказать, что на S-этапе реализуется случайное

«встряхивание» исходной выборки, о котором говорилось выше.)

На втором этапе (*M-этапе*), этапе *максимизации*, в соответствии с формулами (10) и (11) вычисляются оценки максимального правдоподобия компонент параметра θ:

$$p_i^{(m+1)} = \frac{\upsilon_i^{(m+1)}}{n}\,; \tag{12}$$

$$t_i^{(m+1)} = \arg\max_t L_i\left(t; K_i^{(m+1)}\right), i = 1, \ldots, k\,. \tag{13}$$

Наконец, на третьем этапе (*E-этапе*), переназначаются вероятности g_{ij}. Название этого этапа восходит к слову $Expectation$. Это обусловлено тем, что если $\vec{Y}_j^{(m+1)} = \left(Y_{1j}^{(m+1)}, Y_{2j}^{(m+1)}, \ldots, Y_{kj}^{(m+1)}\right)$ — это случайный вектор, реализацией которого является вектор $\vec{y}_j^{(m+1)}$, а $\vec{X} = (X_1, \ldots, X_n)$ — это случайный вектор, реализацией которого является выборка $\mathbf{x} = (x_1, \ldots, x_n)$, то по определению

$$g_{ij}^{(m+1)} = E_{\theta^{(m+1)}}\left(Y_{ij}^{(m+1)} | X_j\right)$$

($Y_{ij}^{(m+1)}$ — это индикатор (случайного) события $\left\{X_j \in K_i^{(m+1)}\right\}$, а математическое ожидание индикатора случайного события равно вероятности этого события). При известном значении $X_j = x_j$ имеем

$$g_{ij}^{(m+1)} = \frac{p_i^{(m+1)}\psi_i\left(x_j; t_i^{(m+1)}\right)}{\sum\limits_{r=1}^{k} p_r^{(m+1)}\psi_r\left(x_j; t_r^{(m+1)}\right)}\,. \tag{14}$$

Для случая смеси нормальных распределений, в которой

$$\psi_i(x; t_i) = \frac{1}{\sigma_i}\,\phi\left(\frac{x - a_i}{\sigma_i}\right),\ x \in \mathbb{R}\,,$$

SEM-алгоритм выглядит так. Соотношение (12) остается без изменений, соотношение (13) трансформируется в два соотношения:

$$a_i^{(m+1)} = \frac{1}{\upsilon_i^{(m+1)}}\sum_{j=1}^{n} y_{ij}^{(m+1)} x_j = \frac{1}{\upsilon_i^{(m+1)}}\sum_{j:\, x_j \in K_i^{(m+1)}} x_j\,; \tag{15}$$

$$\sigma_i^{(m+1)} = \left[\frac{1}{\upsilon_i^{(m+1)}}\sum_{j=1}^{n} y_{ij}^{(m+1)}\left(x_j - a_i^{(m+1)}\right)^2\right]^{1/2} = \left[\frac{1}{\upsilon_i^{(m+1)}}\sum_{j:\, x_j \in K_i^{(m+1)}}\left(x_j - a_i^{(m+1)}\right)^2\right]^{1/2}.$$

Соотношение же (14) примет вид

$$g_{ij}^{(m+1)} = \frac{\dfrac{p_i^{(m+1)}}{\sigma_i^{(m+1)}}\exp\left\{-\dfrac{1}{2}\left(\dfrac{x_j - a_i^{(m+1)}}{\sigma_i^{(m+1)}}\right)^2\right\}}{\sum\limits_{r=1}^{k}\dfrac{p_r^{(m+1)}}{\sigma_r^{(m+1)}}\exp\left\{-\dfrac{1}{2}\left(\dfrac{x_j - a_r^{(m+1)}}{\sigma_r^{(m+1)}}\right)^2\right\}}\,.$$

6 Медианная модификация SEM-алгоритма

Так как на каждой итерации SEM-алгоритма в каждый из кластеров $K_1^{(m+1)}, \ldots, K_k^{(m+1)}$ могут попасть «лишние» наблюдения, фактически распределенные в соответствии с другими компонентами смеси, то можно рассмотреть устойчивые медианные модификации.

Медианные модификации SEM-алгоритма определяются следующим образом. Упорядочим элементы выборки $\mathbf{x} = (x_1, \ldots, x_n)$, попавшие в кластер $K_i^{(m+1)}$, по неубыванию. Полученный в результате набор обозначим

$$K_i^{(m+1)} = \left\{x_{i,1}^{(m+1)}, \ldots, x_{i,\upsilon_i}^{(m+1)}\right\}.$$

В этом случае для оценки параметров используется выборочная медиана для каждого кластера.

В случае смеси нормальных компонент вместо (15) можно использовать более устойчивую оценку

$$a_i^{(m+1)} = \begin{cases} \dfrac{1}{2}\left(x_{i,\upsilon_i^{(m+1)}/2}^{(m+1)} + x_{i,\upsilon_i^{(m+1)}/2+1}^{(m+1)}\right), \\ \qquad \text{если } \upsilon_i^{(m+1)} \text{ — четное}\,; \\ x_{i,\left[\upsilon_i^{(m+1)}/2\right]+1}^{(m+1)}, \\ \qquad \text{если } \upsilon_i^{(m+1)} \text{ — нечетное}\,, \end{cases}$$

где символ $[z]$ обозначает целую часть числа z. Другими словами, в качестве оценки параметра a_i на $(m+1)$-й итерации SEM-алгоритма можно использовать выборочную медиану кластера $K_i^{(m+1)}$.

В качестве оценки параметра σ_i на $(m+1)$-й итерации SEM-алгоритма можно взять

$$\sigma_i^{(m+1)} = \sqrt{\frac{\pi}{2}} \cdot S_i^{(m+1)},$$

где $S_i^{(m+1)}$ — выборочное среднее абсолютное отклонение, вычисленное для кластера $K_i^{(m+1)}$:

$$S_i^{(m+1)} = \frac{1}{\upsilon_i^{(m+1)}} \sum_{j=1}^{\upsilon_i^{(m+1)}} \left| x_{i,j}^{(m+1)} - a_i^{(m+1)} \right|.$$

Таким образом, SEM-алгоритм и его медианная модификация представляют собой методы для оценивания неизвестных параметров компонент смеси без каких-либо дополнительных предположений об этих параметрах (например, предположения о равенстве нулю параметра a_i для каждой компоненты).

7 Выбор точности приближений

Свойства SEM-алгоритма были подвергнуты исследованию в [9, 10]. В частности, в этих работах для многих достаточно общих конкретных случаев отмечено, что построенная SEM-алгоритмом последовательность $\left\{\theta^{(m)}\right\}_{m\geq 1}$, вообще говоря, не сходится с вероятностью единица, но образует цепь Маркова, которая при некоторых дополнительных условиях регулярности довольно быстро сходится к стационарному распределению. Стационарность достигается после довольно продолжительного периода «приработки» алгоритма. При этом получаемые с помощью SEM-алгоритма оценки параметров смеси являются асимптотически несмещенными в том смысле, что оценка максимального правдоподобия параметров смеси является асимптотически эквивалентной математическому ожиданию $\theta^{(m)}$ относительно стационарного распределения. Поэтому в качестве «окончательной» оценки $\widetilde{\theta}^{(m)}$ параметра θ после m итераций SEM-алгоритма в упомянутых работах предлагается использовать «выборочное среднее»

$$\widetilde{\theta}^{(m)} = \widetilde{\theta}^{(m)}(m_0) = \frac{1}{m-m_0} \sum_{r=m_0+1}^{m} \theta^{(m)},$$

где m_0 — настолько большое число, что при $r > m_0$ цепь Маркова $\theta^{(r)}$ близка к стационарному режиму.

Многочисленные реализации SEM-алгоритма показали, что он работает относительно быстро по сравнению с другими методами, результаты его работы практически не зависят от начального приближения, он позволяет избегать выхода на неустойчивые локальные максимумы анализируемой функции правдоподобия за счет постоянного случайного «встряхивания» выборки и, более того, как правило, приводит к глобальному максимуму этой функции. Кроме того, SEM-алгоритм легко модифицировать с целью отыскания числа k компонент смеси, если оно заранее неизвестно (отметим, что в реализованных алгоритмах фактически используется данная идея: если в кластер попадает менее двух элементов, он считается пустым и удаляется).

8 Декомпозиция волатильности с помощью метода скользящего разделения смесей

Возможности описанных выше медианных модификаций EM- и SEM-алгоритмов для разделения конечных смесей нормальных законов будут проиллюстрированы на примере решения задачи декомпозиции волатильности (т. е. задачи разложения волатильности на компоненты) некоторых финансовых индексов.

Статистическое оценивание параметров конечных смесей нормальных законов является ядром метода скользящего разделения смесей (СРС-метода), предназначенного для исследования стохастической структуры хаотических процессов и, в частности, для исследования волатильности финансовых индексов и других показателей.

Теоретические основы СРС-метода можно кратко описать следующим образом (см. [11, 12]).

(1) Асимптотический подход, основанный на предельных теоремах для обобщенных дважды стохастических пуассоновских процессов как моделей неоднородных хаотических случайных блужданий, естественно приводит к заключению о том, что аппроксимации для распределений (логарифмов) приращений процессов эволюции финансовых индексов на интервалах времени умеренной длины следует искать в виде общих сдвиг-масштабных смесей нормальных законов, в которых смешивающий закон определяется накопленной (интегральной) интенсивностью потоков соответствующих информативных событий (элементарных скачков, «тиков»).

(2) Проблема статистической реконструкции распределений приращений упомянутых процессов (или их логарифмов) сводится к задаче статистического оценивания смешивающего

распределения, которое является параметром этой задачи.

(3) В самой общей постановке задача статистического оценивания смешивающего распределения является некорректной, так как общие сдвиг-масштабные смеси нормальных законов не являются идентифицируемыми. Таким образом, в рамках общего принципа регуляризации некорректных задач исходная проблема заменяется задачей отыскания решения, наиболее близкого к истинному в классе конечных дискретных сдвиг-масштабных смесей нормальных законов. Эта «редуцированная» задача уже является корректной и имеет единственное решение, так как семейство конечных дискретных сдвиг-масштабных смесей нормальных законов идентифицируемо. Поскольку сдвиг-масштабные смеси нормальных законов обладают свойством устойчивости относительно смешивающего закона, эта замена оправданна и регулярна. При этом, зная оценки устойчивости, можно вычислить погрешности, образующиеся при замене исходной задачи редуцированной. При упомянутой регуляризации происходит автоматическое выделение типичных или более-менее устойчивых структур в эволюции рассматриваемых сложных систем.

(4) Представление распределений (логарифмов) приращений процессов эволюции финансовых индексов в виде конечных сдвиг-масштабных смесей нормальных законов естественно приводит к многомерной интерпретации волатильности рассматриваемого процесса и к возможности разложения волатильности на динамическую и диффузионную компоненты. Действительно, если функция распределения логарифмического приращения Z некоторого финансового индекса имеет вид (1), то для нее справедливо представление

$$F(x) = \mathsf{P}(Z < x) = \sum_{j=1}^{k} p_j \Phi\left(\frac{x - a_j}{\sigma_j}\right) = \mathsf{E}\Phi\left(\frac{x - V}{U}\right),$$

где пара случайных величин U, V имеет дискретное распределение

$$\mathsf{P}((U,V) = (\sigma_j, a_j)) = p_j\,, \quad j = 1, \ldots, k\,.$$

Так что, как продемонстрировано в книге [12],

$$\mathsf{D}Z = \mathsf{D}V + \mathsf{E}U^2\,, \qquad (16)$$

причем

$$\mathsf{D}V = \sum_{j=1}^{k} (a_j - \overline{a})^2 p_j\,, \quad \mathsf{E}U^2 = \sum_{j=1}^{k} p_j \sigma_j^2\,, \quad (17)$$

где

$$\overline{a} = \sum_{j=1}^{k} a_j p_j\,.$$

Волатильность индекса естественно отождествить с величиной $\mathsf{D}Z$ (или $\sqrt{\mathsf{D}Z}$). При этом первое выражение в (17) зависит только от весов p_j и параметров положения (сдвига) a_j компонент, и потому характеризует ту часть волатильности, которая обусловлена наличием локальных трендов, т. е. «динамическую» компоненту волатильности, тогда как второе выражение в (17) зависит только от весов p_j и параметров масштаба («коэффициентов диффузии») σ_j компонент и потому характеризует «чисто диффузионную» компоненту волатильности.

Если вспомнить традиционное одномерное представление о волатильности как о стандартном отклонении приращения процесса, то можно заметить, что разложение (16) уточняет это представление: волатильность процесса представляет собой корень квадратный из суммы двух компонент, первая из которых является характеристикой разбросанности локальных трендов, а вторая характеризует диффузию процесса. Если локальные тренды отсутствуют, то классическая волатильность равна корню квадратному из взвешенной суммы квадратов волатильностей компонент, причем веса компонент показывают важность соответствующей диффузионной компоненты.

(5) Статистические закономерности поведения рассматриваемых процессов, формализованные в пункте (1), изменяются во времени, результатом чего является отсутствие *универсального* смешивающего закона. Таким образом, чтобы изучить динамику изменения статистических закономерностей в поведении исследуемого хаотического процесса, задача статистического разделения конечных смесей нормальных законов должна быть последовательно решена на интервалах времени, постоянно сдвигающихся в направлении «астрономического» времени. Тем самым параметры смесей (параметры сдвига (дрейфа), масштаба (диффузии), а также соответствующие веса) оцениваются как функции времени. При этом естественно возникают задачи, связанные как с

выбором подходящих методов оценивания параметров сдвига и масштаба, так и с выбором оптимальных параметров вычислительных процедур, реализующих эти методы: начального приближения, ширины скользящего интервала (окна), правила остановки и др.

(6) Наконец, для адекватной интерпретации результатов и для идентификации феноменологически выделенных (статистически оцененных) компонент, т. е. для адекватного сопоставления статистически оцененных компонент с реальными процессами или явлениями, необходимо из многих возможных моделей выбрать наиболее адекватную, например проверить, является выделенная динамическая компонента волатильности статистически значимой или нет.

9 Диффузионный спектр и предполагаемый диффузионный спектр

Сказанное в пункте (4) предыдущего раздела позволяет предложить новую точку зрения на природу волатильности.

Чтобы дать более полный ответ на вопрос о том, что такое волатильность, помня тем не менее о том, что среди финансовых аналитиков «волатильность» — это ныне скорее обыденное понятие, нежели математический термин, необходимо заметить, что волатильность процесса (т. е. его изменчивость) обусловлена как минимум двумя типами факторов. Первый тип факторов может быть условно назван *динамическим*. Влияние факторов такого типа проявляется в том, что процесс изменяется из-за наличия некоторого тренда или комбинации, взаимодействия трендов, отражающих интересы некоторых (нескольких) групп участников рынка (простейшим примером таких групп являются «быки», играющие на повышение, и «медведи», играющие на понижение). Упрощенная физическая интерпретация действия этой группы факторов может быть проиллюстрирована примером движения, скажем, щепки в горной реке, где имеются ярко выраженные течения или комбинации течений, перемещающие щепку. Следует отметить, что обычно при описании методов анализа эволюции финансовых индексов понятие тренда в некотором смысле противопоставляется понятию волатильности и оба эти понятия изучаются раздельно. Согласно сказанному в пункте (4) предыдущего раздела, CPC-метод показывает, что на самом деле между этими понятиями имеется глубокая и непростая взаимосвязь, которую, к счастью, можно довольно разумно описать, используя аппарат теории вероятностей и математической статистики.

Факторы второго типа могут быть названы *стохастическими* или *диффузионными*. Число факторов, влияющих на финансовые рынки (параметры и интересы участников, новости и т. п.), огромно, так что на практике нереально учесть влияние каждого из них в отдельности. Физическим примером действия факторов диффузионного типа может служить суммарное воздействие молекул среды на частицу, испытывающую броуновское (тепловое) движение.

Конечно, эта классификация факторов скорее условна, нежели строго определена. Однако CPC-метод спонтанно и довольно естественно выделяет факторы указанных типов и автоматически дает возможность проследить их взаимосвязь (см. соотношения (16) и (17)). В соответствии со сказанным выше в рамках основной модели, используемой в данной работе, волатильность довольно естественно раскладывается на *динамическую* и *диффузионную* составляющие. В свою очередь, диффузионная составляющая раскладывается на несколько разных компонент, каждая из которых имеет свое собственное происхождение. Таким образом, в данной работе рассматривается метод статистического разложения волатильности *скалярного* процесса в нетривиальную и существенно *многомерную* информативную картину. Наблюдаемое значение финансового индекса в каждый момент времени является интегральным (суммарным, усредненным) результатом взаимодействия интересов и стратегий участников рынка. Подобно тому, как Исаак Ньютон с помощью призмы разложил белый свет на составляющие его цвета радуги, с помощью CPC-метода наблюдаемый интегральный индекс, характеризующий состояние рынка, можно в определенном смысле разложить на составляющие его компоненты, отражающие текущие интересы и стратегии характерных групп участников рынка.

Как будет показано на примере исследования конкретных временны́х рядов, описывающих эволюцию реальных финансовых индексов, динамическая и диффузионные составляющие вносят в итоговую волатильность примерно одинаковый вклад, причем в разные моменты времени доминировать могут разные компоненты.

Если модели типа масштабных смесей нормальных законов оказываются адекватными для распределений логарифмических приращений финансовых индексов, то смешивающее распределение можно интерпретировать как *диффузионный спектр* волатильности. Если такие модели в целом оказываются неадекватными по какому-либо критерию (например, адекватной является сдвиг-масштабная

смесь или вообще иная модель, не имеющая вид смеси нормальных законов), то смешивающее распределение в наиболее правдоподобной модели типа чисто масштабной смеси нормальных законов можно интерпретировать как *предполагаемый диффузионный спектр* волатильности. В последней ситуации (возможное) наличие динамической (трендовой) составляющей волатильности игнорируется и все параметры положения a_j компонент смеси полагаются равными нулю.

Такой подход, в определенном смысле упрощающий изучаемую ситуацию и, возможно, ее искажающий, тем не менее может оказаться полезным, поскольку модели, имеющие вид чисто масштабных смесей, и модели типа сдвиг-масштабных смесей по-разному реагируют на их «уточнение» с помощью добавления дополнительных компонент. Для чисто масштабных смесей наблюдается эффект «насыщения», когда введение новых компонент сверх какого-то уровня (обычно четыре–пять компонент) практически не меняет модель, в то время как модели типа сдвиг-масштабных смесей при добавлении новых компонент изменяются довольно существенно, превращаясь в своего рода аналоги ядерных оценок плотности. Эффект «насыщения» ставит естественный предел степени детализации модели, имеющей вид чисто масштабной смеси нормальных законов.

При этом некоторые компоненты предполагаемого диффузионного спектра могут быть интерпретированы как компоненты шума, которым объявляется невязка между моделью, имеющей вид чисто масштабной смеси нормальных законов, и «истинной» (адекватной) моделью, возможно, имеющей более сложный вид. В частности, если «истинная» модель имеет вид сдвиг-масштабной смеси нормальных законов, то к «зашумляющей» невязке при таком подходе относится динамическая компонента волатильности.

Критерии адекватности или неадекватности моделей весьма относительны. Например, известно довольно много информационных критериев выбора моделей типа критерия Акаике, и в силу самого́ их определения они могут приводить к разным решениям относительно адекватности или неадекватности чисто масшабно-смешанных моделей. Поэтому понятие предполагаемого диффузионного спектра оказывается полезным в любом случае.

Будучи смешивающим распределением, диффузионный спектр может оказаться как непрерывным (абсолютно непрерывным), так и дискретным. Диффузионный спектр (предполагаемый диффузионный спектр) характеризует распределение волатильности (отождествляемой, возможно, с многомерным коэффициентом диффузии) по уровням в каждый момент времени. Тем самым изменение диффузионного спектра во времени характеризует перераспределение волатильности по уровням. При этом множество допустимых уровней волатильности может быть как континуальным (что соответствует непрерывному предполагаемому диффузионному спектру), так и счетным (что соответствует дискретному предполагаемому диффузионному спектру).

Наиболее приемлемым с точки зрения возможности удобной интерпретации результатов является дискретный (конечный) предполагаемый диффузионный спектр.

Рассматриваемые далее конкретные примеры проиллюстрируют сказанное.

10 Применение медианных модификаций EM-алгоритма для разделения конечных смесей нормальных законов к декомпозиции волатильности конкретных финансовых индексов

В качестве исходных данных использовались минутные логарифмические приращения четырех биржевых индексов: AMEX, CAC 40, Nasdaq 100 и Nikkei. При этом ширина скользящего окна n взята равной 300 отсчетам (что соответствует пяти часам). Задача оценивания параметров смесей последовательно решается для каждого положения скользящего окна по выборке (отрезку исходного ряда), соответствующей данному положению окна. С целью получения довольно точных результатов использовался весьма строгий критерий остановки итерационных процедур, согласно которому евклидово расстояние между векторами значений оцениваемых параметров на последовательных итерациях должно быть меньше 10^{-8}. Число возможных компонент смеси полагалось равным шести.

Результаты представлены на рис. 1–8. На каждом графике горизонтальная ось — это ось времени (каждая точка на горизонтальной оси соответствует конкретному значению правого конца скользящего интервала времени, по которому (т. е. по наблюдениям, попавшим в который) вычисляются оценки параметров. Вертикальная ось — это ось значений параметров $a_j = a_j(t)$ для портретов динамических (трендовых) компонент волатильности или $\sigma_j = \sigma_j(t)$ для портретов диффузионных компонент

волатильности. Веса компонент смеси, соответствующих конкретным значениям параметров a_j и σ_j, показаны оттенками серого цвета. Чем линия темнее, тем вес больше.

Для сравнения на рисунках также представлены результаты решения аналогичной задачи с помощью EM-алгоритма. Этот алгоритм реализован программой ZHPlot, разработанной Ю. В. Жуковым.

На рис. 1 и 2 представлены результаты анализа волатильности индекса AMEX с помощью CPC-метода. По рис. 1 видно, что игнорирование фактически ненулевой динамической (трендовой) составляющей волатильности приводит к тому, что портрет предполагаемой диффузионной волатильности (*а*) оказывается довольно «мохнатым», на нем трудно выделить явные компоненты. Однако как только наличие нетривиальных динамических компонент признается и они автоматически оцениваются, картина становится намного более четкой. При этом как в диффузионной, так и в динамической составляющих волатильности выделяются по две компоненты, причем в каждой паре одна имеет явно выраженный периодический характер. Этот эффект значительно более наглядно проявляется на рис. 2. Если в портрете предполагаемой диффузионной волатильности (*а*) присутствуют элементы «хаотических шумов», хотя и меньших по сравнению с портретом, получаемым обычным EM-алгоритмом (рис. 1, *а*), то рис. 1, *б* и *в* объясняют наличие этого «шума»: он оказывается обусловленным наличием явно выраженных трендовых составляющих. При этом обе пары диффузионных и динамических компонент выделяются медианными модификациями EM-алгоритма намного более четко, нежели обычным EM-алгоритмом.

Точно такой же эффект наблюдается на рис. 3 и 4, где приведены результаты анализа волатильности индекса CAC 40 с помощью CPC-метода с использованием обычного EM-алгоритма (см. рис. 3) и медианной модификации EM-алгоритма (см. рис. 4). При этом на рис. 3 с трудом угадываются две компоненты предполагаемой диффузионной волатильности, которые довольно хорошо выделяются (с небольшими зашумлениями) медианной версией EM-алгоритма на рис. 4.

Рисунки 5 и 6 великолепно иллюстрируют как разницу между предполагаемой диффузионной волатильностью и самой диффузионной волатильностью, так и неоспоримые преимущества медианных версий EM-алгоритма перед обычным EM-алгоритмом при их использовании для разделения смесей нормальных законов. На рис. 6 очень хорошо видно, что «мнимые» компоненты предполагаемой диффузионной волатильности, «зашумляющие» верхний рисунок, полностью обусловлены наличием динамической компоненты волатильности: на рис. 6, *б* в отличие от рис. 6, *а*, присутствует лишь одна компонента.

На рис. 7 и 8 наблюдается картина, аналогичная приведенным на рис. 1–4.

11 Применение медианных модификаций SEM-алгоритма для разделения конечных смесей нормальных законов к декомпозиции волатильности конкретных финансовых индексов

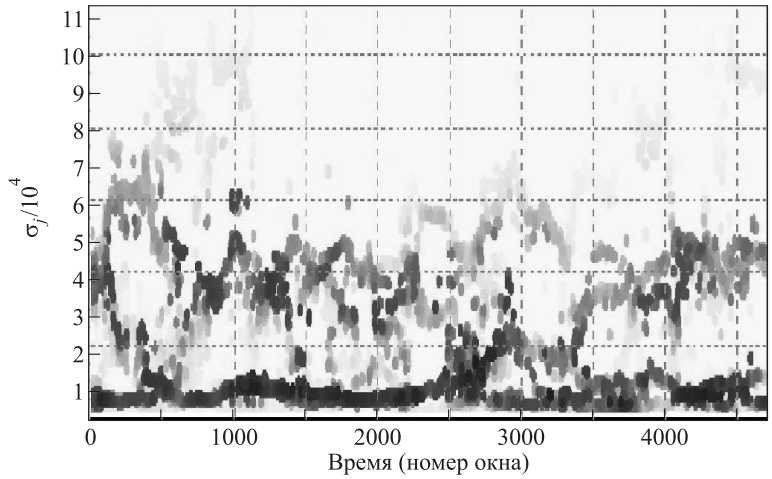
В качестве исходных данных использовались минутные логарифмические приращения биржевых индексов AMEX, CAC 40, Nasdaq 100, Nikkei, S&P500 и PTC, а также контракты на золото и специально смоделированная выборка с известными параметрами. При этом ширина скользящего окна n взята равной 200 отсчетам. Задача оценивания параметров смесей последовательно решалась для каждого положения скользящего окна по выборке (отрезку исходного ряда), соответствующей данному положению окна.

Эмпирически было установлено, что необходимая «приработка» достигается при точности приближения 10^{-5}–10^{-6}. Стоит отметить, что хорошая скорость сходимости достигается и при точности 10^{-12}. Однако в силу некоторых особенностей алгоритма расчеты производились именно для точности ε, равной 10^{-5}–10^{-6}. В качестве критерия останова использовалось соотношение

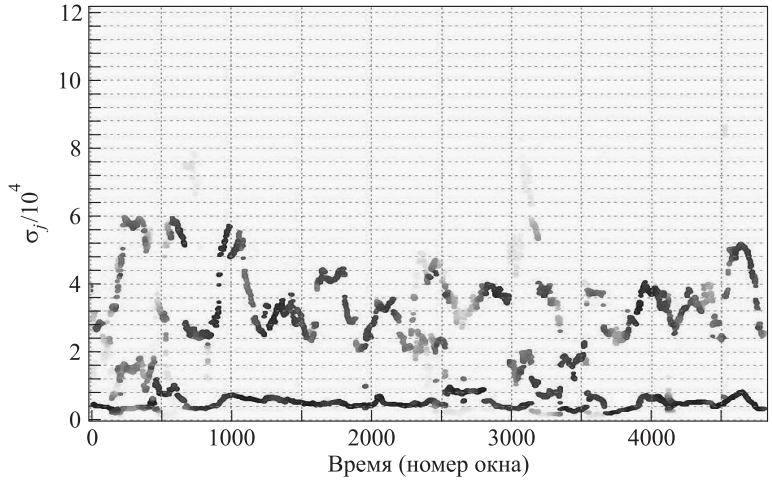
$$\max \left| \theta^{(m)} - \theta^{(m-1)} \right| < \varepsilon \,,$$

где $\theta^{(m)}$ — вектор всех оцениваемых параметров на m-м итерационном шаге, а ε — указанная выше точность. Число возможных компонент смеси полагалось равным шести.

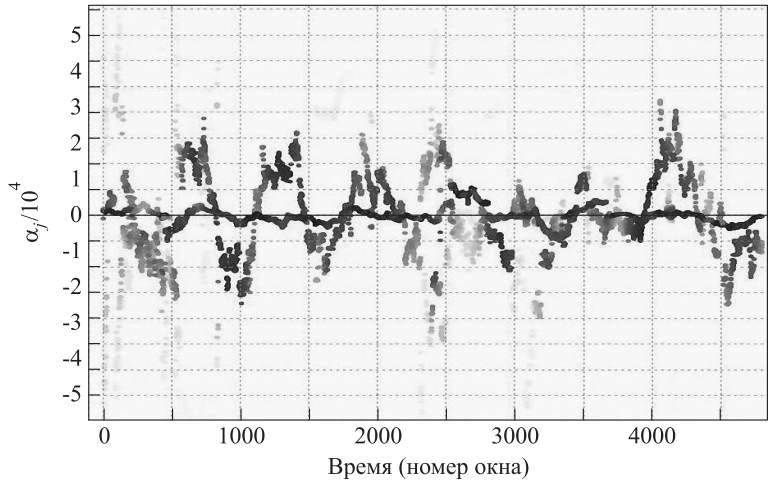
Результаты представлены на рис. 9–20. Как и на предыдущих рисунках, на каждом графике любая точка на горизонтальной оси соответствует конкретному значению правого конца скользящего интервала времени, по которому (т. е. по наблюдениям, попавшим в который) вычисляются оценки параметров. Вертикальная ось — это ось значений параметров $a_j = a_j(t)$ для портретов динамических (трендовых) компонент волатильности или $\sigma_j = \sigma_j(t)$ для портретов диффузионных



(*а*)



(*б*)



(*в*)

Рис. 1 Портреты волатильности индекса AMEX, полученные CPC-методом с использованием EM-алгоритма: предполагаемая диффузионная волатильность (*а*); диффузионная компонента волатильности (*б*); динамическая компонента волатильности (*в*)

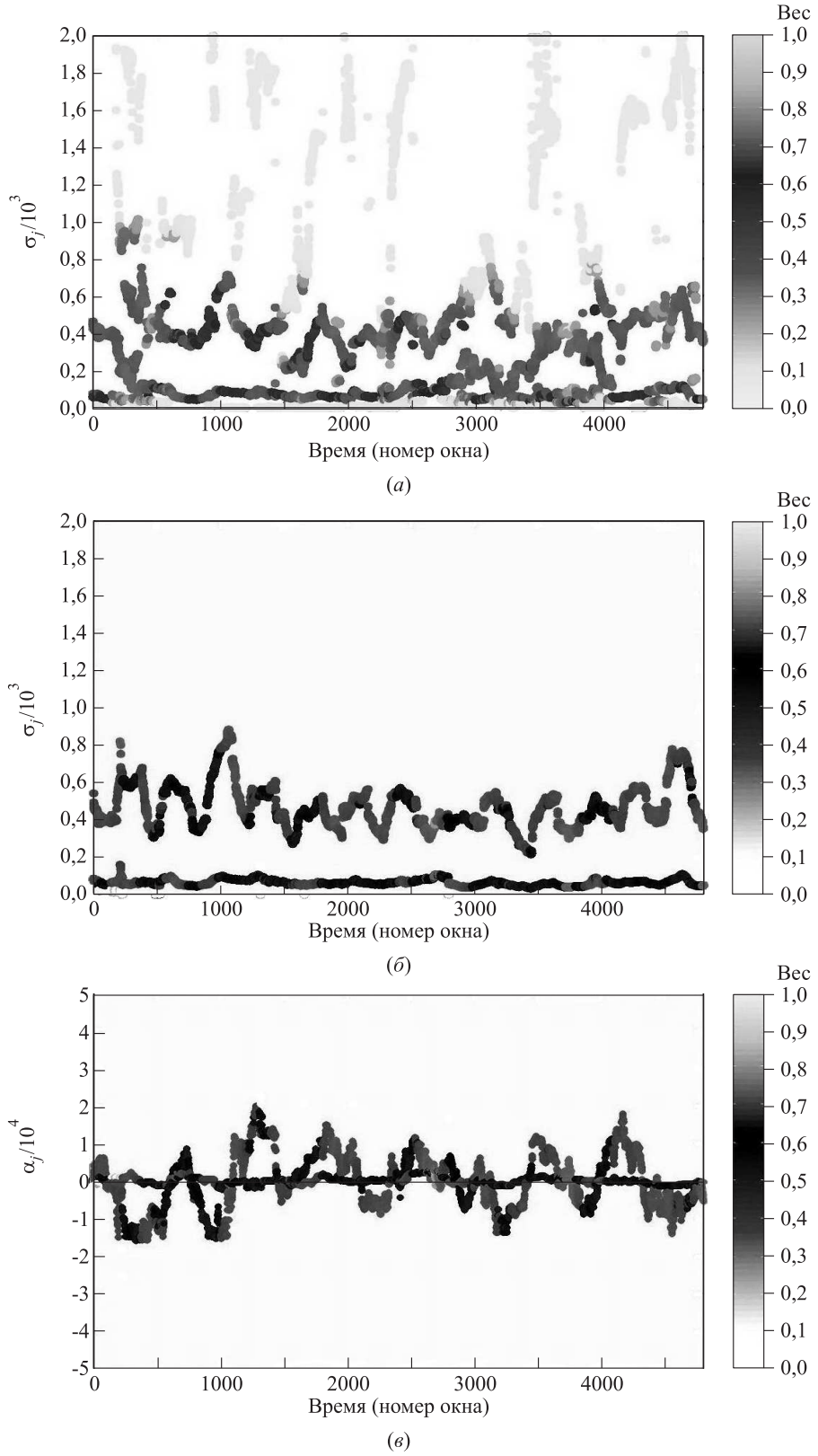


Рис. 2 Портреты волатильности индекса AMEX, полученные CPC-методом с использованием медианной модификации EM-алгоритма: предполагаемая диффузионная волатильность (*а*); диффузионная компонента волатильности (*б*); динамическая компонента волатильности (*в*)

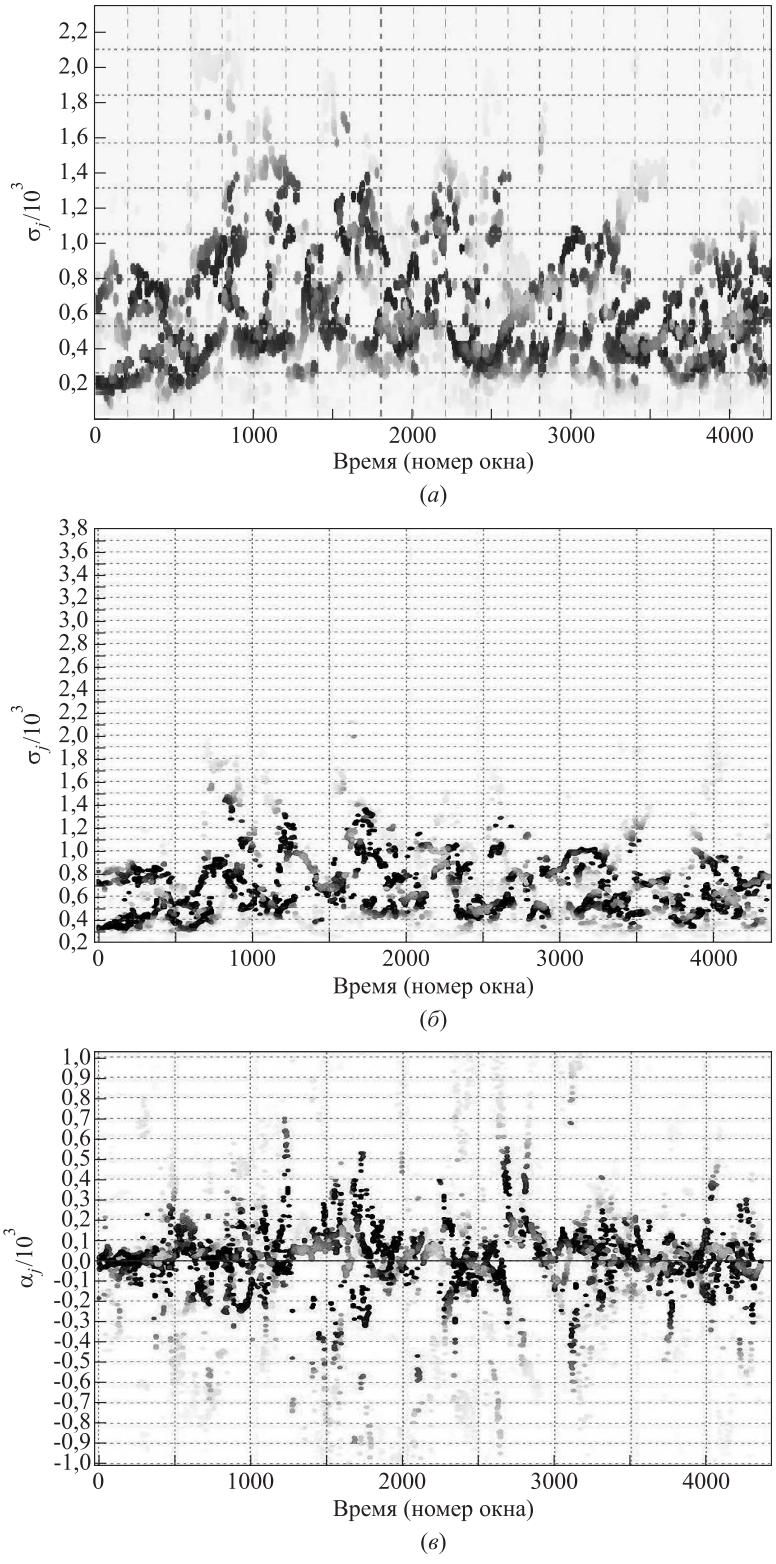


Рис. 3 Портреты волатильности индекса CAC 40, полученные CPC-методом с использованием EM-алгоритма: предполагаемая диффузионная волатильность (*а*); диффузионная компонента волатильности (*б*); динамическая компонента волатильности (*в*)

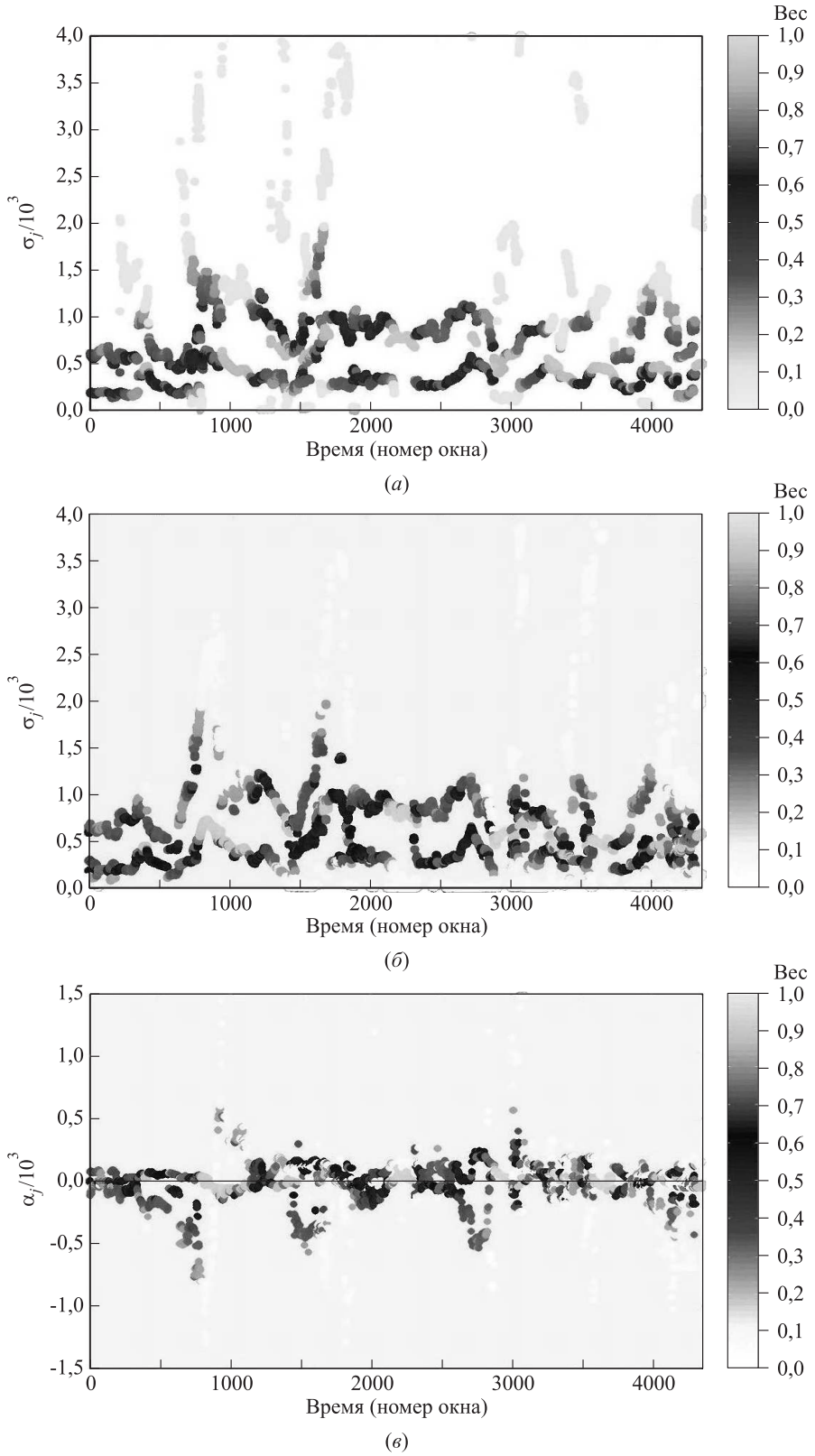


Рис. 4 Портреты волатильности индекса CAC 40, полученные CPC-методом с использованием медианной модификации EM-алгоритма: предполагаемая диффузионная волатильность (*а*); диффузионная компонента волатильности (*б*); динамическая компонента волатильности (*в*)

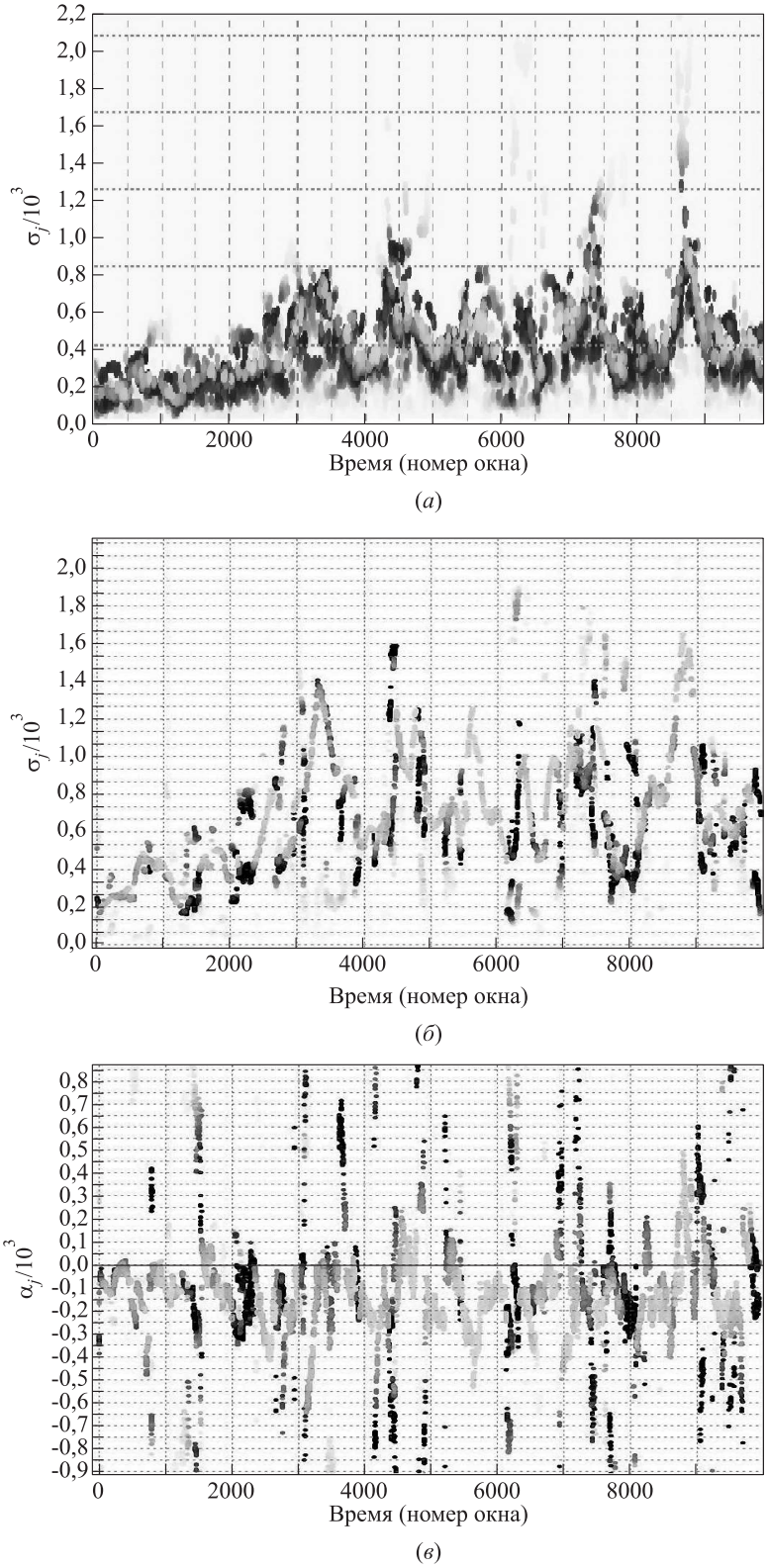


Рис. 5 Портреты волатильности индекса Nasdaq 100, полученные CPC-методом с использованием EM-алгоритма: предполагаемая диффузионная волатильность (*а*); диффузионная компонента волатильности (*б*); динамическая компонента волатильности (*в*)

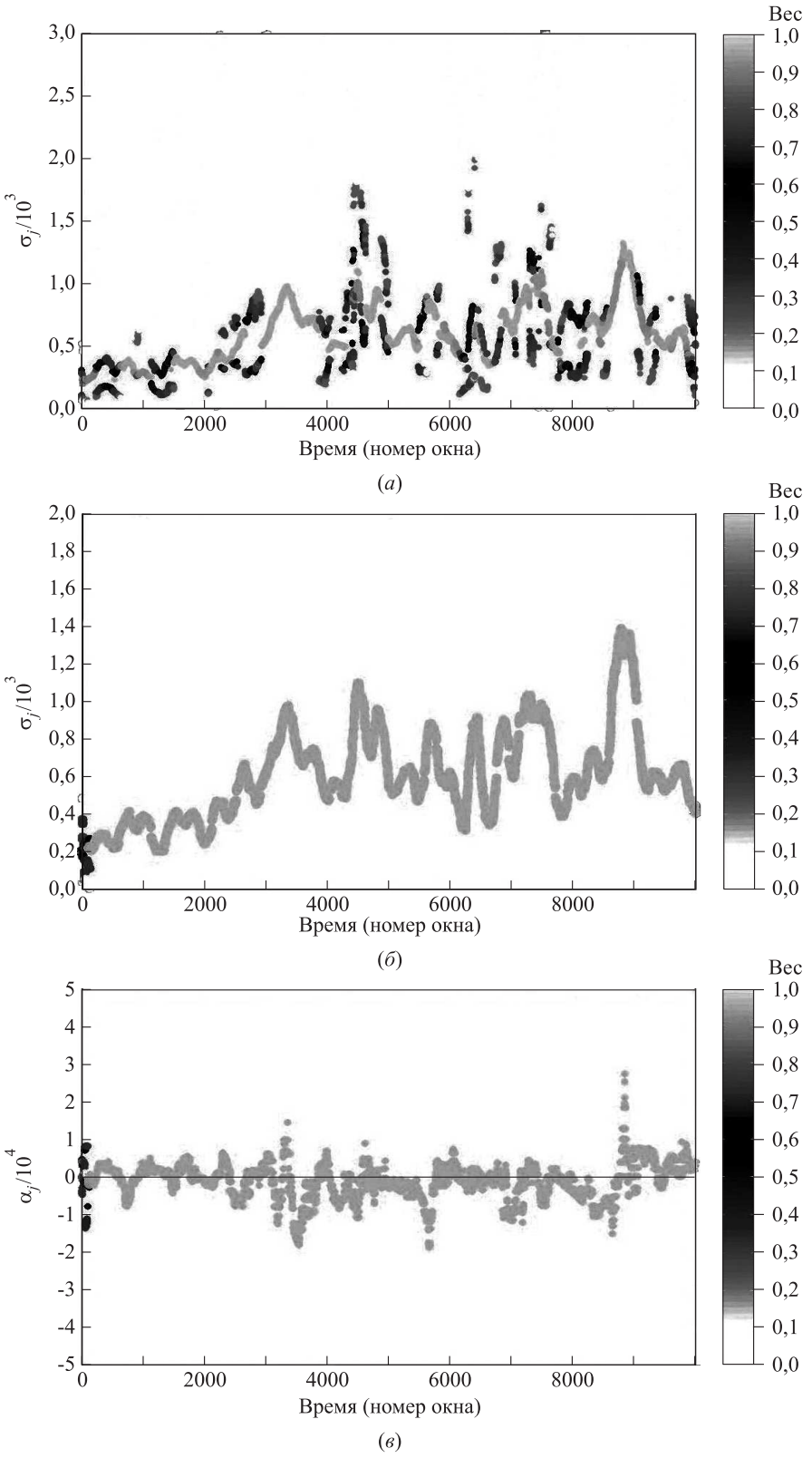


Рис. 6 Портреты волатильности индекса Nasdaq 100, полученные CPC-методом с использованием медианной модификации EM-алгоритма: предполагаемая диффузионная волатильность (*а*); диффузионная компонента волатильности (*б*); динамическая компонента волатильности (*в*)

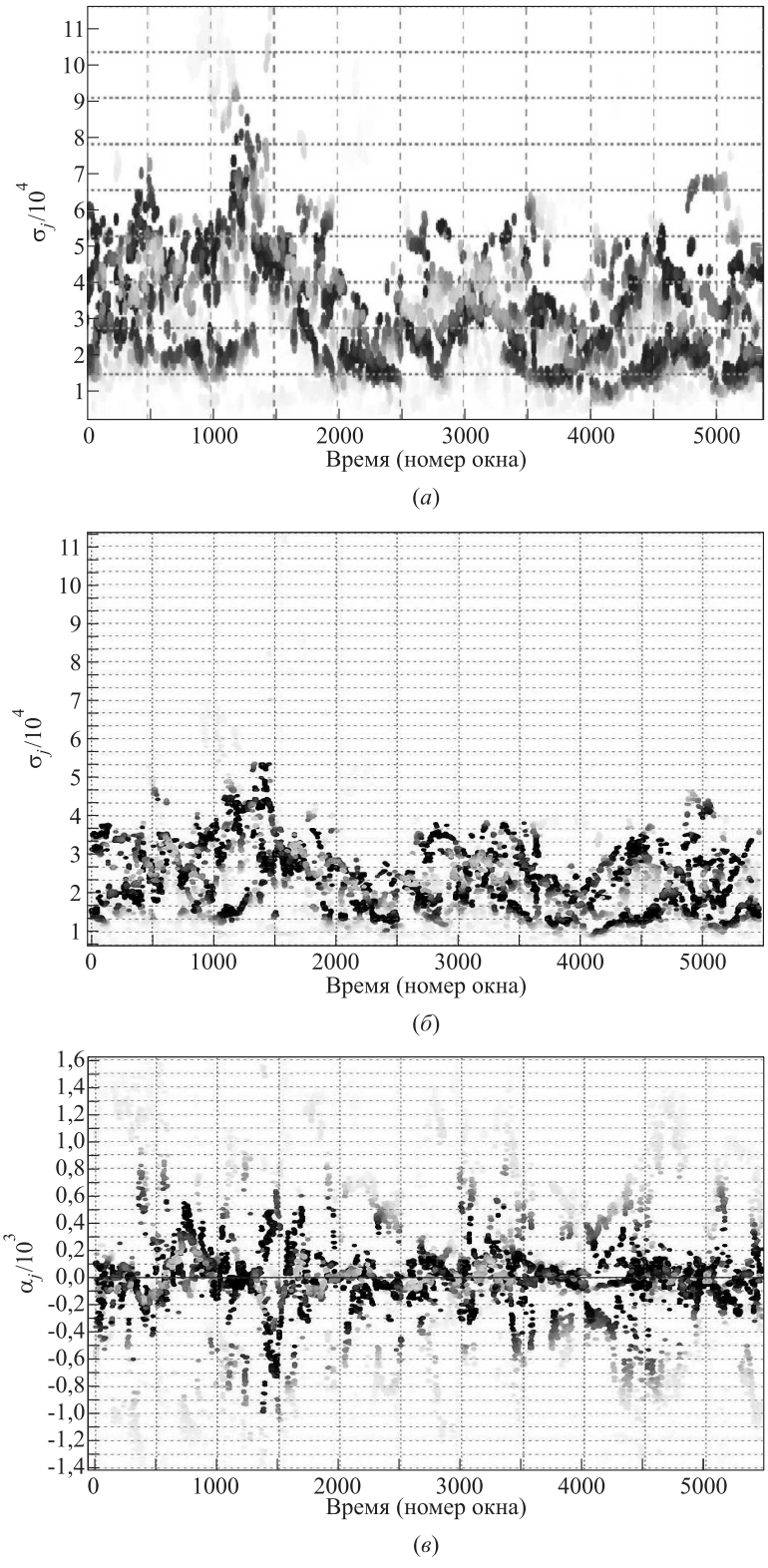


Рис. 7 Портреты волатильности индекса Nikkei, полученные CPC-методом с использованием EM-алгоритма: предполагаемая диффузионная волатильность (*а*); диффузионная компонента волатильности (*б*); динамическая компонента волатильности (*в*)

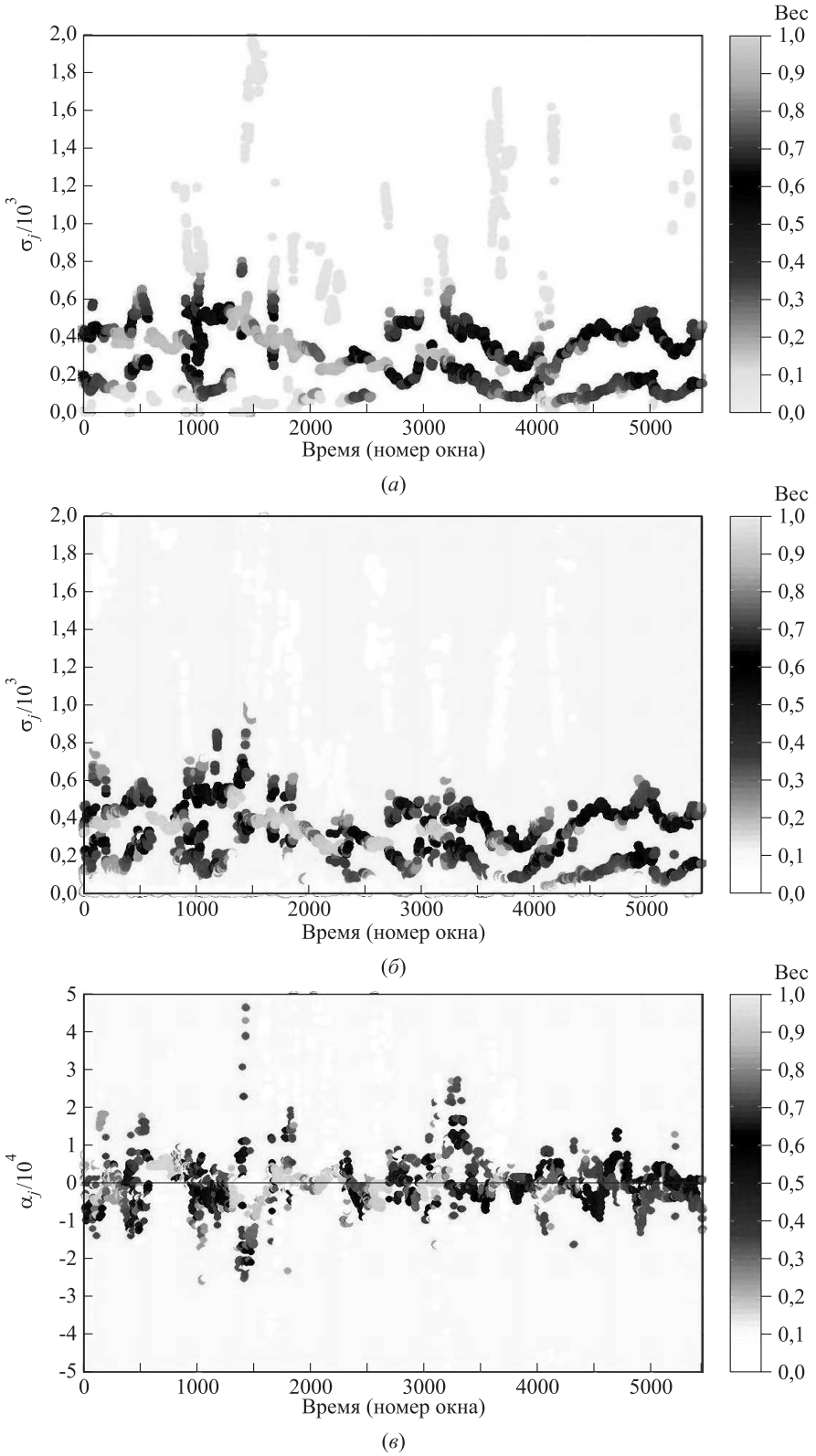


Рис. 8 Портреты волатильности индекса Nikkei, полученные CPC-методом с использованием медианной модификации EM-алгоритма: предполагаемая диффузионная волатильность (*а*); диффузионная компонента волатильности (*б*); динамическая компонента волатильности (*в*)

компонент волатильности. Веса компонент смеси, соответствующих конкретным значениям параметров a_j и σ_j показаны оттенками серого цвета. Чем линия темнее, тем вес больше.

Для сравнения на рисунках также представлены результаты решения аналогичной задачи с помощью EM-алгоритма (при точности приближения, равной 10^{-8}).

На рис. 9–12 представлены результаты анализа диффузионной волатильности индексов, рассмотренных в разд. 10. Так как используется меньшая точность итерационных приближений, результаты получаются более зашумленными, однако характерное поведение компонент (а также их число) сохраняется.

Рассмотрим ряд других индексов. На рис. 13 представлен анализ диффузионной компоненты волатильности для индекса NASDAQ за три торговых дня.

На рис. 13, *а* изображены результаты применения EM-алгоритма (окно 200, точность 10^{-8}). Результат достаточно сильно «зашумлен», однако можно говорить о наличии одной компоненты смеси. На рис. 13, *б* приводятся результаты применения SEM-алгоритма (окно 200, точность 10^{-6}). На данном графике четко просматривается наличие одной компоненты с большим весом (весьма близким к 1), а также наличие низковолатильной компоненты с весом, близким к 0. Отметим наличие изломов на графике компоненты. При этом график рис. 13, *в* результатов применения медианной модификации SEM-алгоритма (окно 200, точность 10^{-6}) получается практически гладким. Приведенный ряд интересен тем, что удалось обнаружить наличие единственной компоненты, которая, пусть и на протяжении не очень длительного периода времени, оказывала определяющее влияние на поведение индекса NASDAQ.

Похожая картина (с соответствующими выводами по каждому методу) наблюдается и на рис. 14 анализа диффузионной компоненты волатильности для индекса S&P500: значительная «зашумленность» результатов EM-алгоритма (окно 200, точность 10^{-8}), большая четкость результатов SEM-алгоритма (окно 200, точность 10^{-6}) и, наконец, большая гладкость и отсутствие ложного дробления компонент на графике результатов применения медианной модификации SEM-алгоритма (окно 200, точность 10^{-6}).

На рис. 15 приведен анализ диффузионной компоненты волатильности для индекса РТС. Отметим значительно более высокую наглядность графика для медианной модификации SEM-алгоритма (окно 200, точность 10^{-5}), на котором все компоненты видны максимально четко.

Как отмечалось выше, SEM-алгоритм и его медианные модификации предназначены для оценки всех параметров смеси без каких-либо предварительных предположений об их значениях. Ранее рассматривались только оценки параметра масштаба. Теперь рассмотрим и динамику параметров сдвига — проанализируем динамические (трендовые) компоненты волатильности.

На рис. 16 приведен результат анализа динамической компоненты волатильности для индекса РТС. Все три сравниваемых метода выделяют похожую картину волатильности, на графике SEM-алгоритма (окно 200, точность 10^{-5}) она видна более четко, а на графике медианной модификации SEM-алгоритма (окно 200, точность 10^{-5}) еще и практически избавлена от шумов. Для данного ряда результаты оценивания динамической компоненты волатильности повторяют результаты, полученные для диффузионной компоненты: лучшая визуализация достигается применением медианной модификации SEM-алгоритма. Однако этот вывод справедлив для динамической компоненты далеко не всегда.

На рис. 17 приведен анализ диффузионной компоненты волатильности для золота. Результаты аналогичны тем, что получены для индексов. Обратим внимание на заметную периодичность компоненты с наибольшим весом. Период составляет около 250 шагов, что согласуется с дневной активностью на рынках (данные минутные, торговый день составляет 480 минут — 240 шагов в пятиминутных данных). Таким образом, SEM-алгоритм и его медианная модификация выявили периодичность в активности торгов (что хорошо согласуется с ситуацией на рынке), при этом видны 7 периодов, данные представлены за 8 торговых дней (но не полных). EM-алгоритм данную периодичность выявить не смог.

В разд. 3 приводились обоснования эффективности использования медианных оценок. Однако было отмечено, что необходимые для этого условия выполняются не всегда. В качестве примера рассмотрим рис. 18, на котором приведены результаты анализа динамической компоненты волатильности для золота.

На рис. 18, *а*, где приведены результаты применения EM-алгоритма (окно 200, точность 10^{-8}), компоненты не выделяются. На рис. 18, *б*, где приведены результаты применения SEM-алгоритма, четко различима трендовая компонента. На рис. 18, *в*, где приведены результаты применения медианной модификации SEM-алгоритма, тренда нет совсем — компоненты преимущественно оцениваются нулем. Обратим внимание на вид результатов медианной модификации SEM-алгоритма:

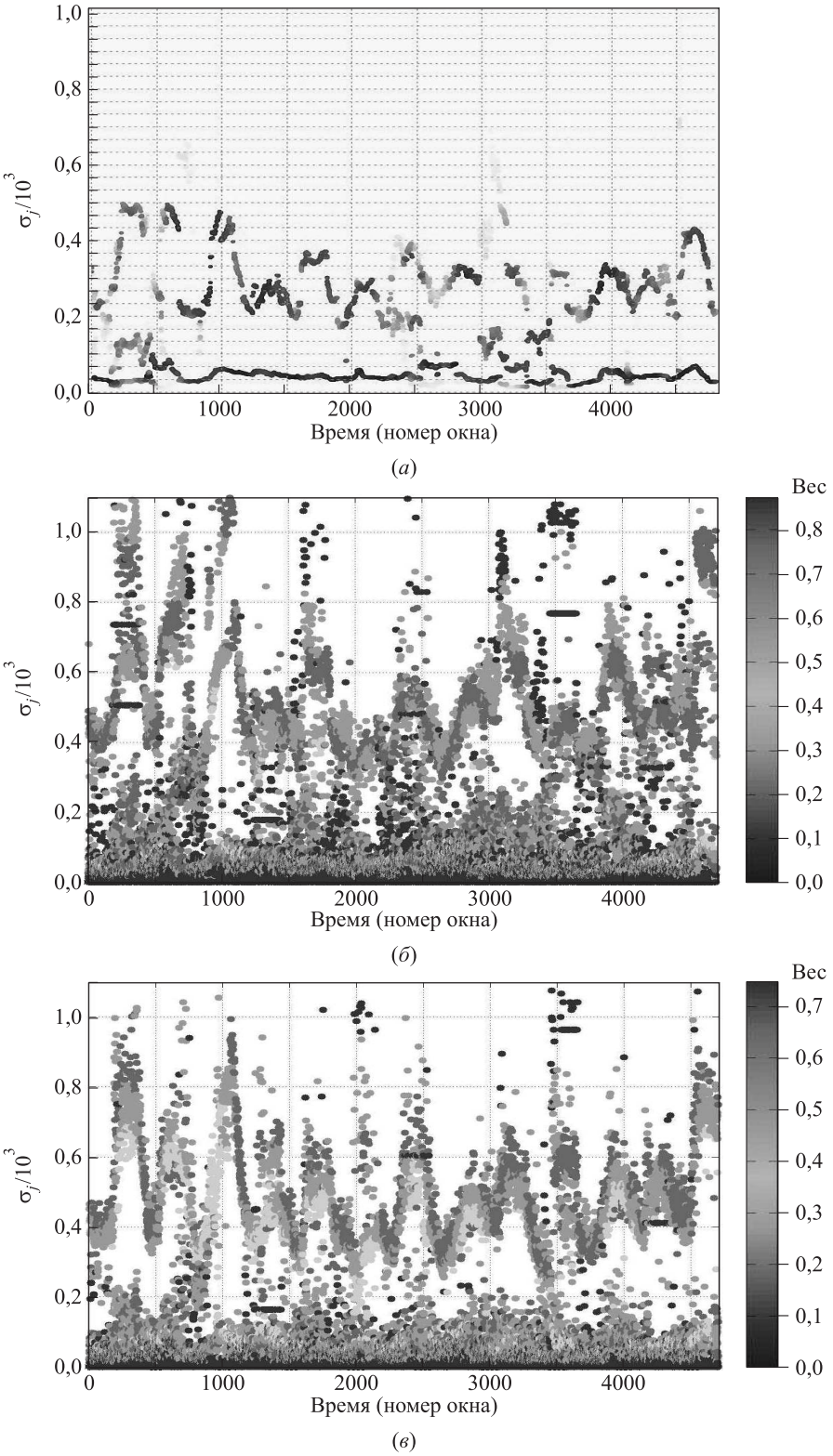


Рис. 9 Портреты диффузионной волатильности индекса AMEX, полученные: EM-алгоритмом (*а*); SEM-алгоритмом (*б*); медианной модификацией SEM-алгоритма (*в*)

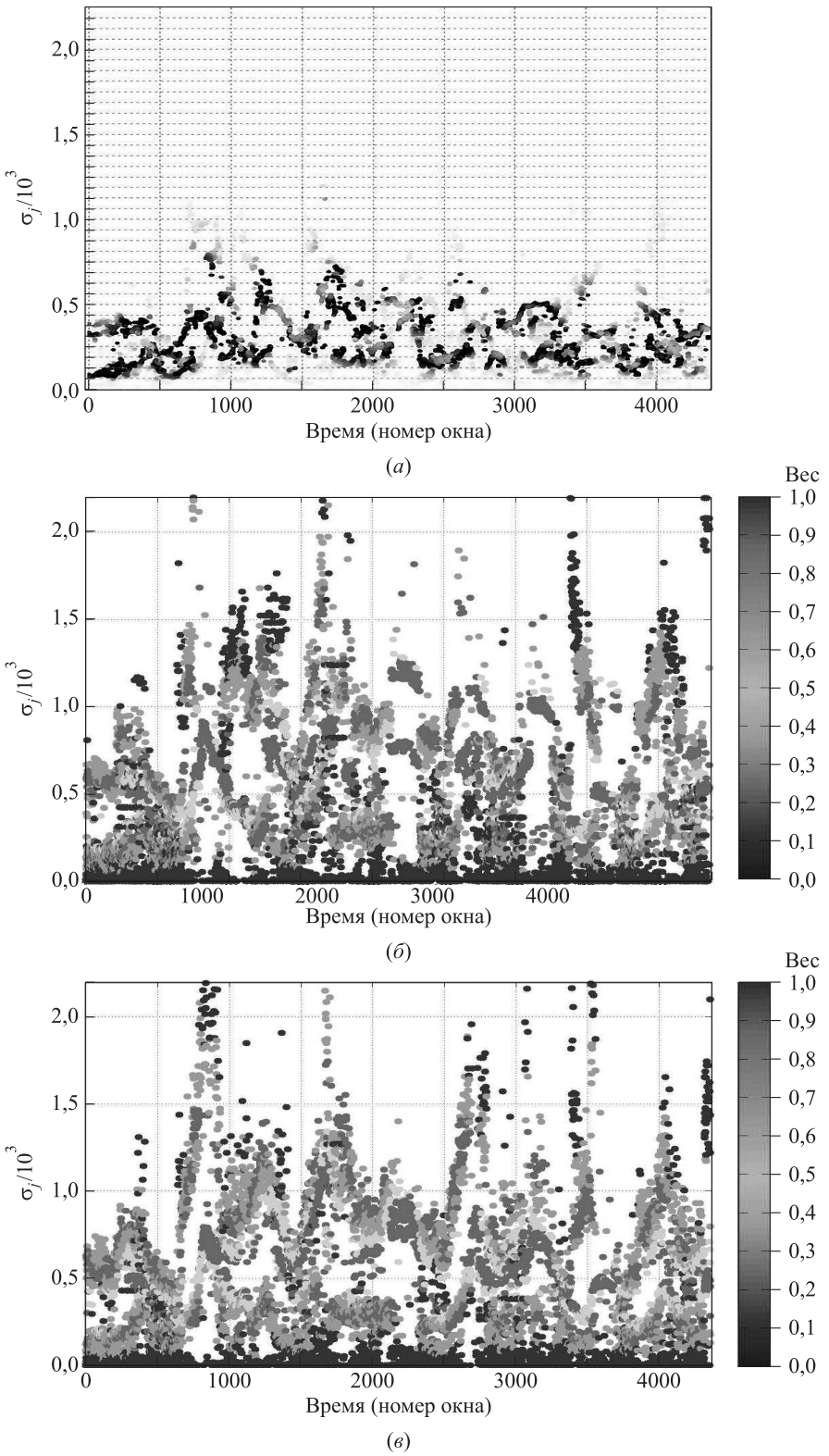


Рис. 10 Портреты диффузионной волатильности индекса CAC 40, полученные: EM-алгоритмом (*а*); SEM-алгоритмом (*б*); медианной модификацией SEM-алгоритма (*в*)

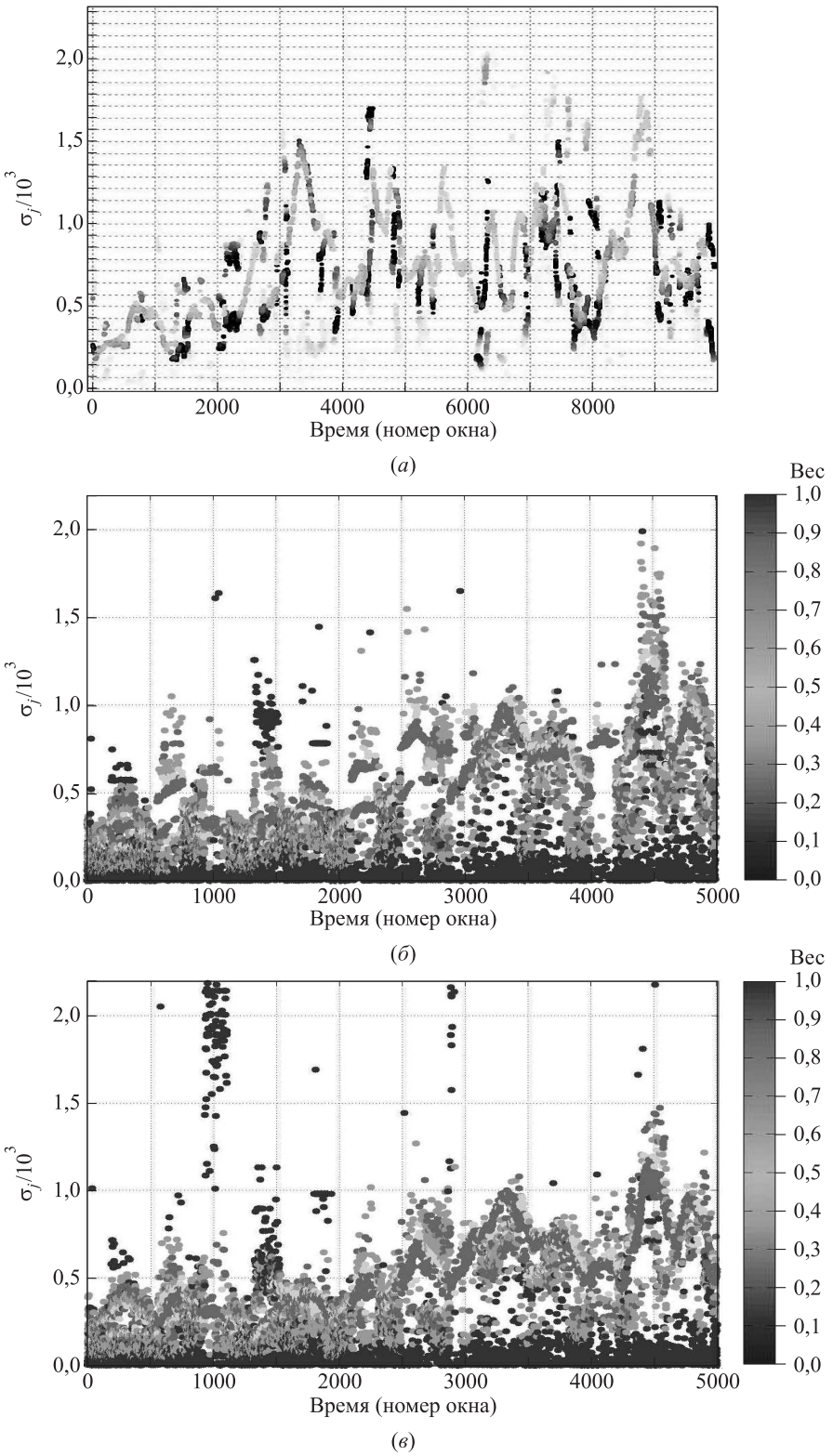


Рис. 11 Портреты диффузионной волатильности индекса Nasdaq 100, полученные: EM-алгоритмом (*а*); SEM-алгоритмом (*б*); медианной модификацией SEM-алгоритма (*в*)

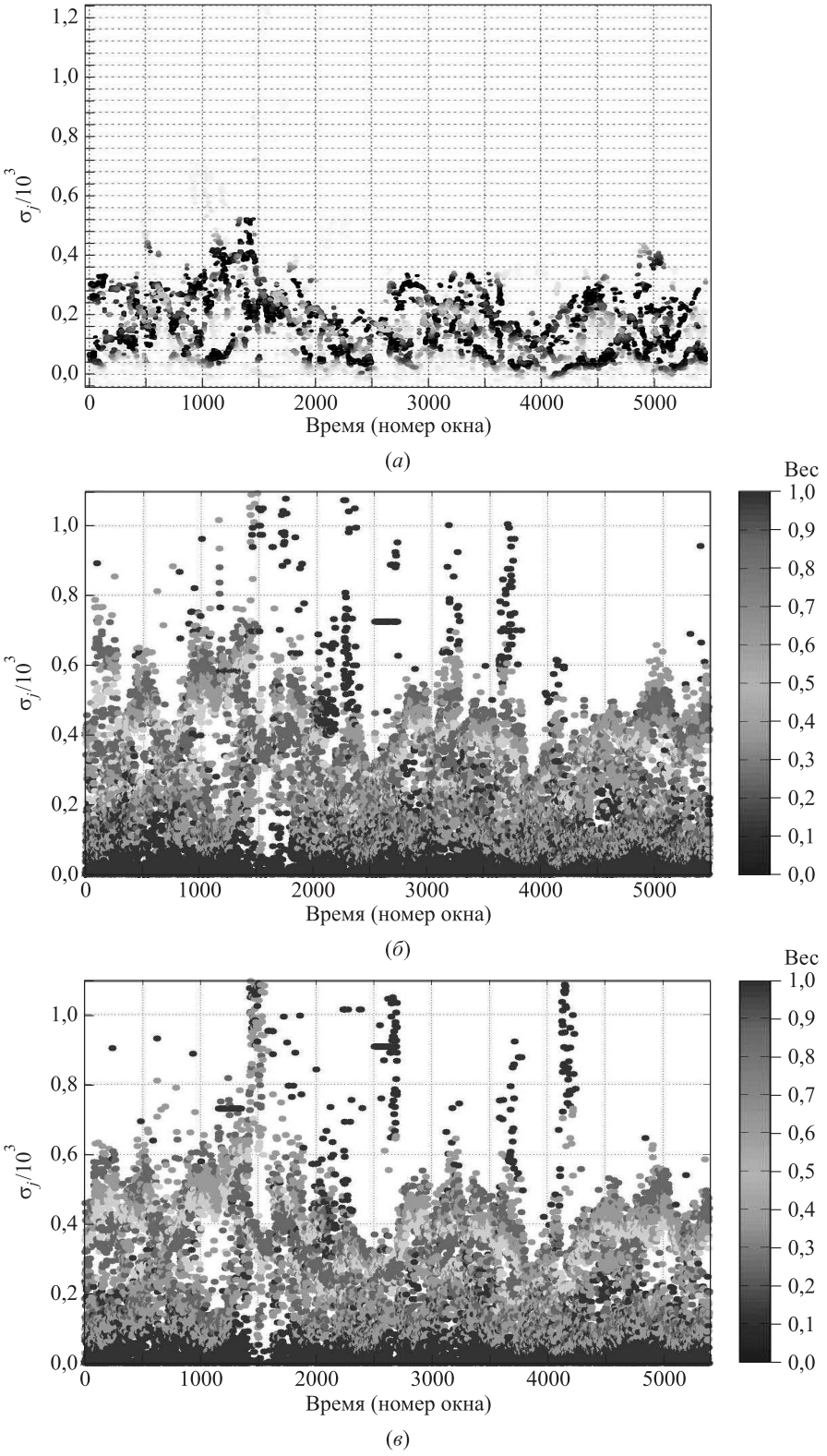


Рис. 12 Портреты диффузионной волатильности индекса Nikkei, полученные: EM-алгоритмом (*а*); SEM-алгоритмом (*б*); медианной модификацией SEM-алгоритма (*в*)

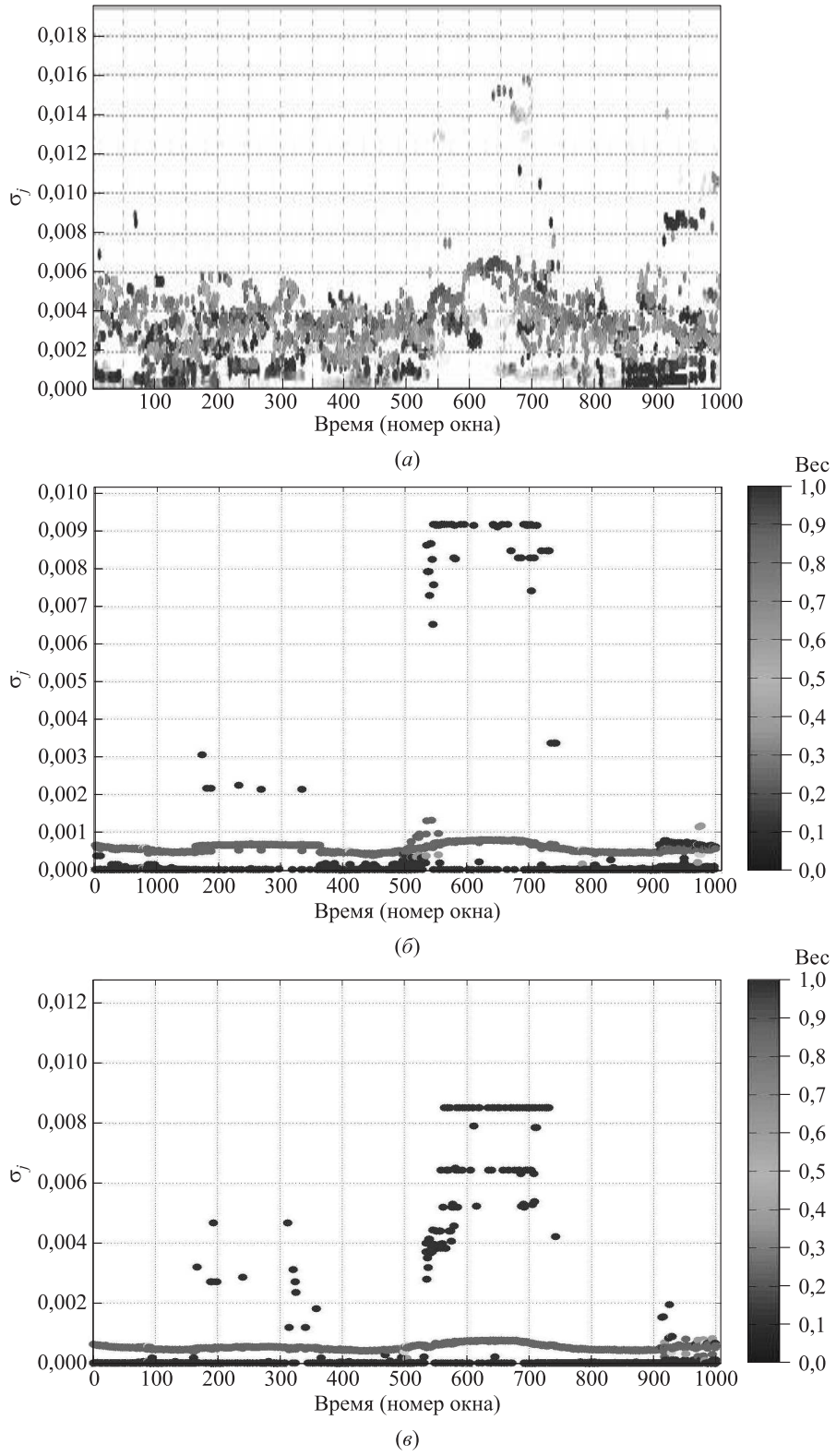


Рис. 13 Портреты диффузионной волатильности индекса NASDAQ, полученные: EM-алгоритмом (*а*); SEM-алгоритмом (*б*); медианной модификацией SEM-алгоритма (*в*)

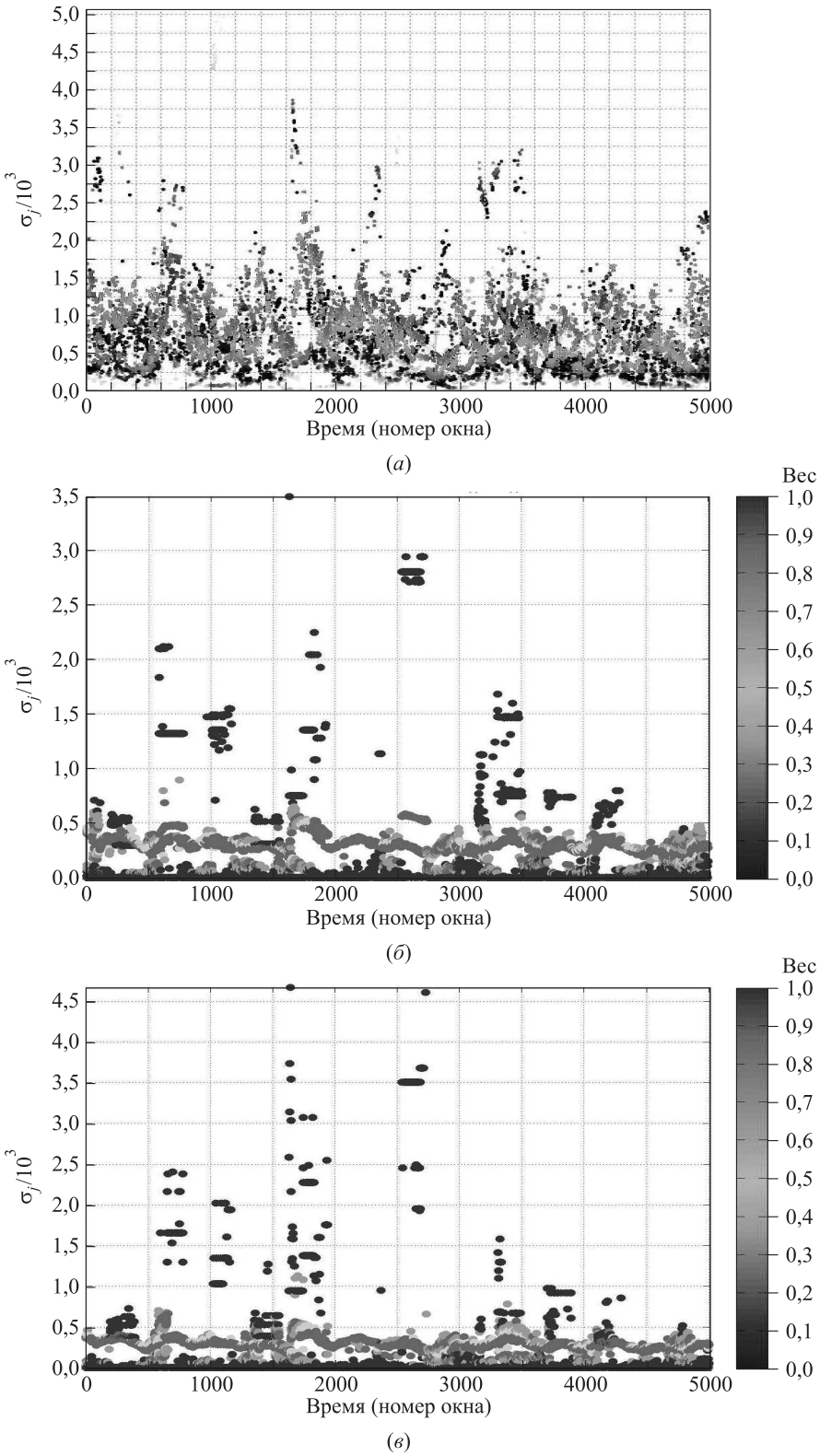


Рис. 14 Портреты диффузионной волатильности индекса S&P500, полученные: EM-алгоритмом (*а*); SEM-алгоритмом (*б*); медианной модификацией SEM-алгоритма (*в*)

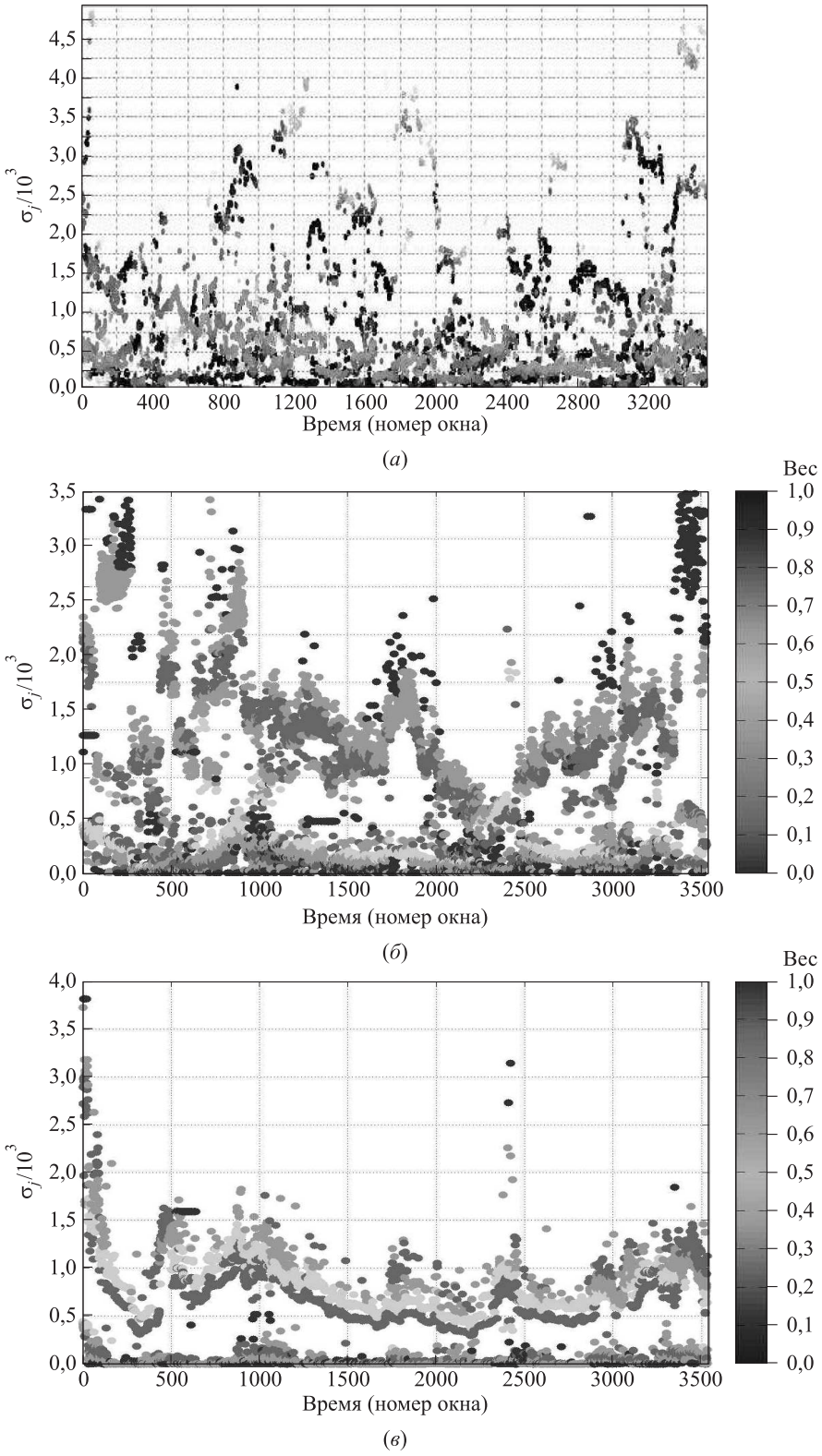


Рис. 15 Портреты диффузионной волатильности индекса РТС, полученные: EM-алгоритмом (*а*); SEM-алгоритмом (*б*); медианной модификацией SEM-алгоритма (*в*)

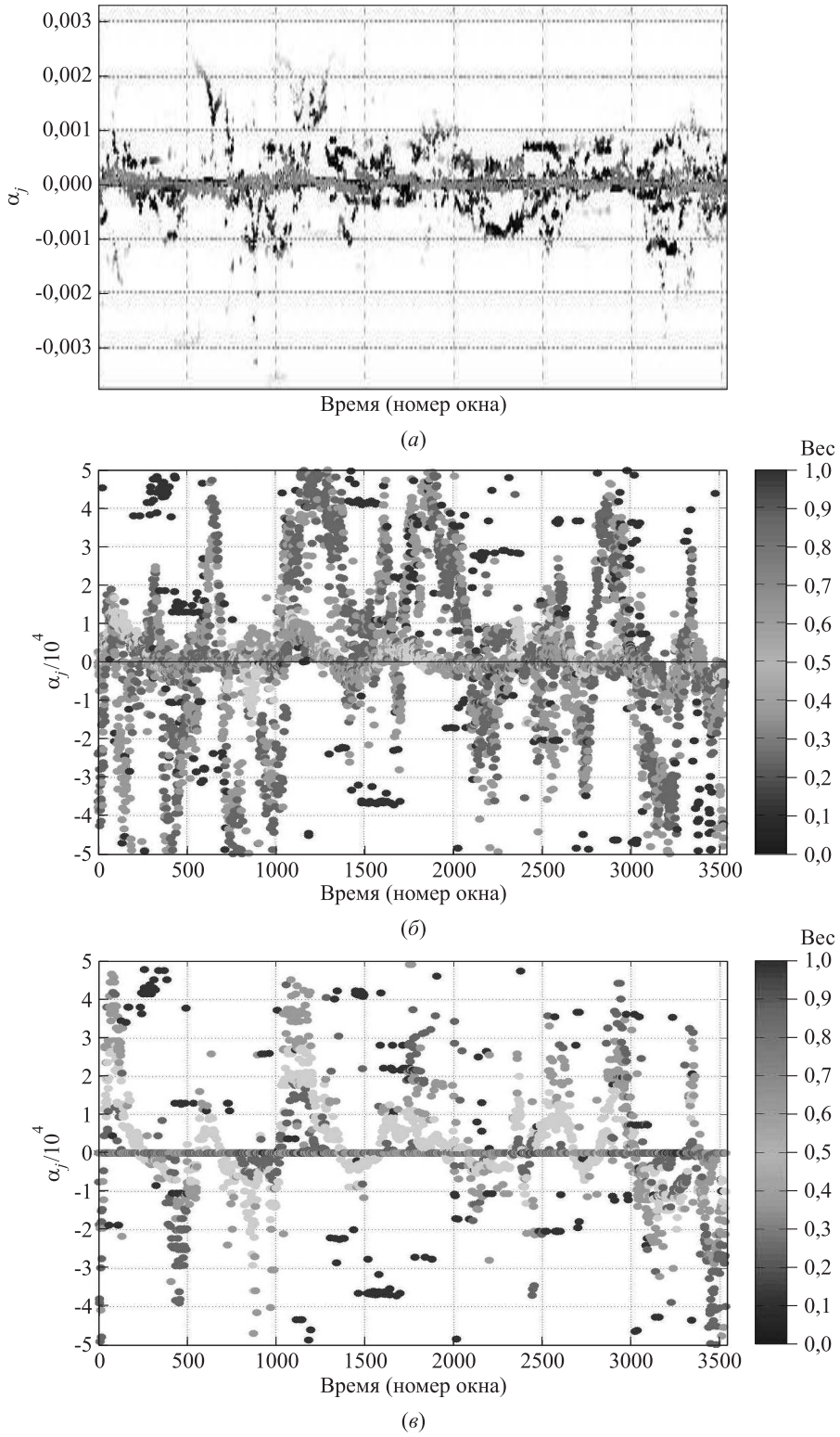


Рис. 16 Портреты динамической волатильности индекса РТС, полученные: EM-алгоритмом (*а*); SEM-алгоритмом (*б*); медианной модификацией SEM-алгоритма (*в*)

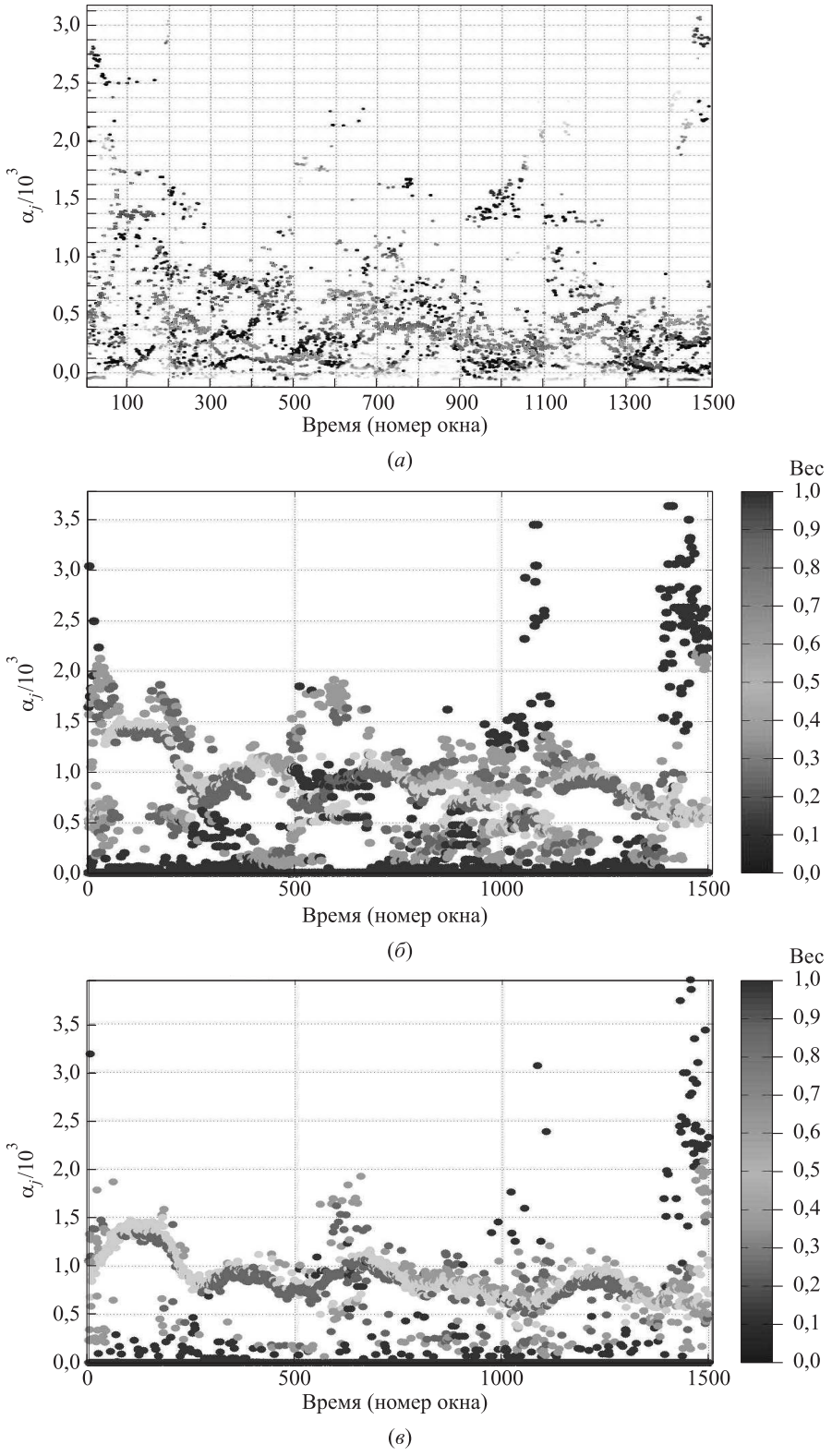


Рис. 17 Портреты динамической волатильности для золота, полученные: EM-алгоритмом (*а*); SEM-алгоритмом (*б*); медианной модификацией SEM-алгоритма (*в*)

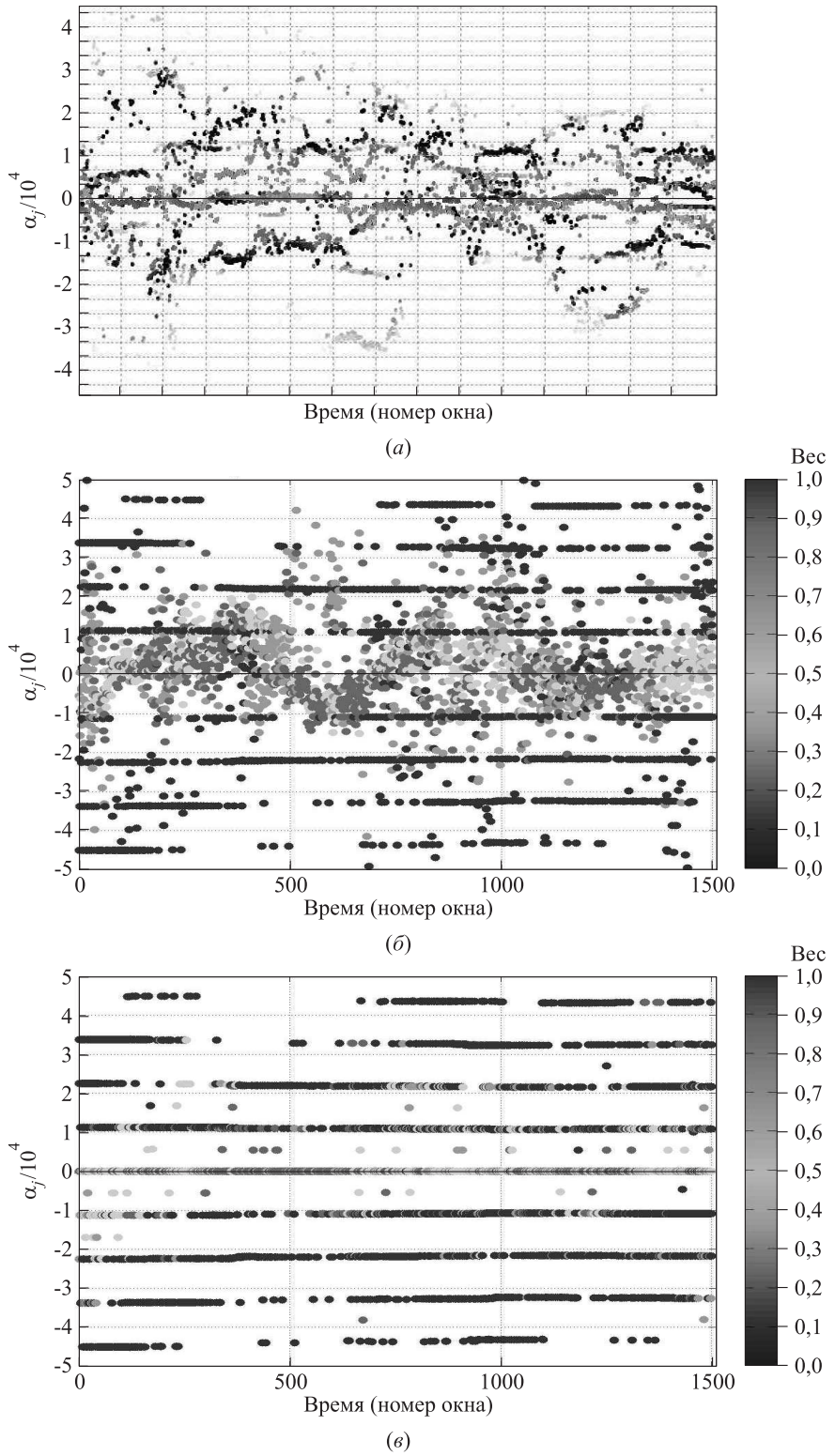


Рис. 18 Портреты динамической волатильности для золота, полученные: EM-алгоритмом (*а*); SEM-алгоритмом (*б*); медианной модификацией SEM-алгоритма (*в*)

график представляет собой набор «полос». Такой вид объясняется тем, что в качестве оценки параметра сдвига используется выборочная медиана. Благодаря тому что она по определению представляет собой либо элемент выборки, либо полусумму двух элементов выборки, на каждом следующем шаге выборка изменяется ровно на один элемент, и поскольку медиана является робастной оценкой, получаются горизонтальные участки оценок параметров, которые на графике представляют собой «полосы». Расстояние между ними объясняется различием в точности, с которой получены оценки (фактически представляющих собой элементы выборки) и с которой заданы исходные данные. Так, в данном примере выборка задана с точностью порядка 10^{-3}, в то время как точность алгоритма составляет 10^{-6}.

Таким образом, получается, что медианная модификация SEM-алгоритма эффективна для оценки диффузионных компонент (результаты почти всегда значительно лучше аналогичных для обычного SEM-алгоритма). Однако при оценке динамической компоненты встречаются ситуации, когда разделение компонент SEM-алгоритмом проводится правильно, но результаты, получаемые медианной модификацией SEM-алгоритма, с трудом поддаются анализу.

Наконец, рассмотрим тестовую выборку. Моделируем выборку из смеси нормальных распределений с весами 1/3, 1/3 и 1/3 с параметрами $(0,(0{,}001)^2)$, $(0,(0{,}005)^2)$ и $(0,(0{,}004)^2)$, всего 1000 элементов. Сравним результаты алгоритмов, примененных к этому ряду. Сначала проанализируем диффузионную компоненту волатильности. На рис. 19, *а* изображены результаты применения EM-алгоритма (окно 200, точность 10^{-6}). Определить число компонент достаточно сложно — возможны две или более компонент. При этом ни одна из дисперсий исходных компонент не оценена правильно. На рис. 19, *б* приводятся результаты применения SEM-алгоритма (окно 200, точность 10^{-6}). Компонента, соответствующая параметру $(0{,}005)^2$, плавно переходит в компоненту, соответствующую $(0{,}004)^2$. Данный алгоритм позволяет визуально обнаружить только две компоненты. А вот на графике рис. 19, *в* результатов применения медианной модификации SEM-алгоритма (окно 200, точность 10^{-6}) компоненты уже четко разделены — видны правильные оценки трех(!) компонент. Итак, EM-алгоритм определил число компонент неправильно, также неверно оценены параметры компонент: как дисперсия, так и математическое ожидание. SEM-алгоритм обнаружил только две компоненты, хотя оценки параметров, по сути, верны для всех трех компонент. Наконец, медианная модификация SEM-алгоритма оценивает параметры правильно и обнаруживает все компоненты. Несмотря на то что оценки параметров при этом весьма близки, компоненты не сливаются в одну.

На рис. 20 приведен анализ динамической компоненты волатильности для тестовой выборки. Видно, что результаты EM-алгоритма сложны для интерпретации. В то же время результаты SEM-алгоритма и его медианной модификации похожи, что говорит о выполнении условий эффективности медианных оценок.

12 Заключение

1. Медианные модификации EM-алгоритма продемонстрировали намного большую пригодность к использованию для решения задачи численного (статистического) разделения смесей нормальных законов по сравнению с обычным EM-алгоритмом. Портреты волатильности финансовых индексов, получаемые с помощью медианных модификаций EM-алгоритма, отличаются большей четкостью, гладкостью и, следовательно, более наглядны и удобны для интерпретации.

2. Статистический анализ данных о поведении финансовых индексов свидетельствует в пользу наличия нетривиальных динамических компонент, возможность существования которых вытекает из базовой модели в рамках метода скользящего разделения смесей.

3. В пользу адекватности базовой модели типа конечной смеси нормальных законов свидетельствует и то, что при максимально возможных шести компонентах смеси практически значимыми оказывались лишь 1–3 компоненты.

4. Результаты, получаемые медианными модификациями SEM-алгоритма, не требуют никаких дополнительных предположений о начальных приближениях (что существенно используется в данной реализации медианного EM-алгоритма). При этом медианные модификации SEM-алгоритма позволяют получать весьма хорошо интерпретируемые результаты с меньшей точностью приближений, что позволяет затрачивать меньшее время на вычисления. Медианный EM-алгоритм без учета дополнительных предположений зачастую дает худшие результаты (даже на большей точности), сравнимые с результатами применения стандартного EM-алгоритма.

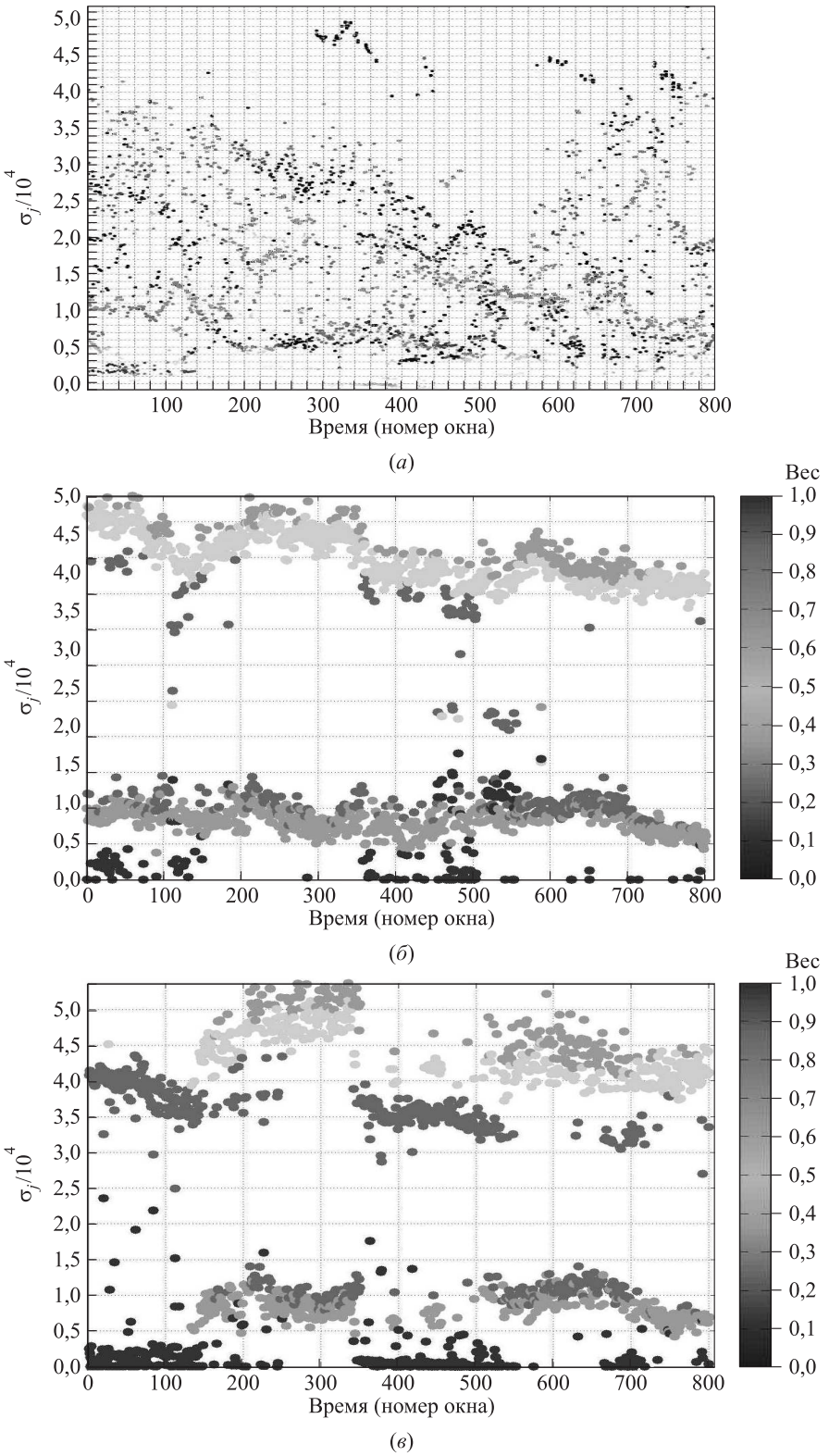


Рис. 19 Портреты диффузионной волатильности тестовой выборки, полученные: EM-алгоритмом (*а*); SEM-алгоритмом (*б*); медианной модификацией SEM-алгоритма (*в*)

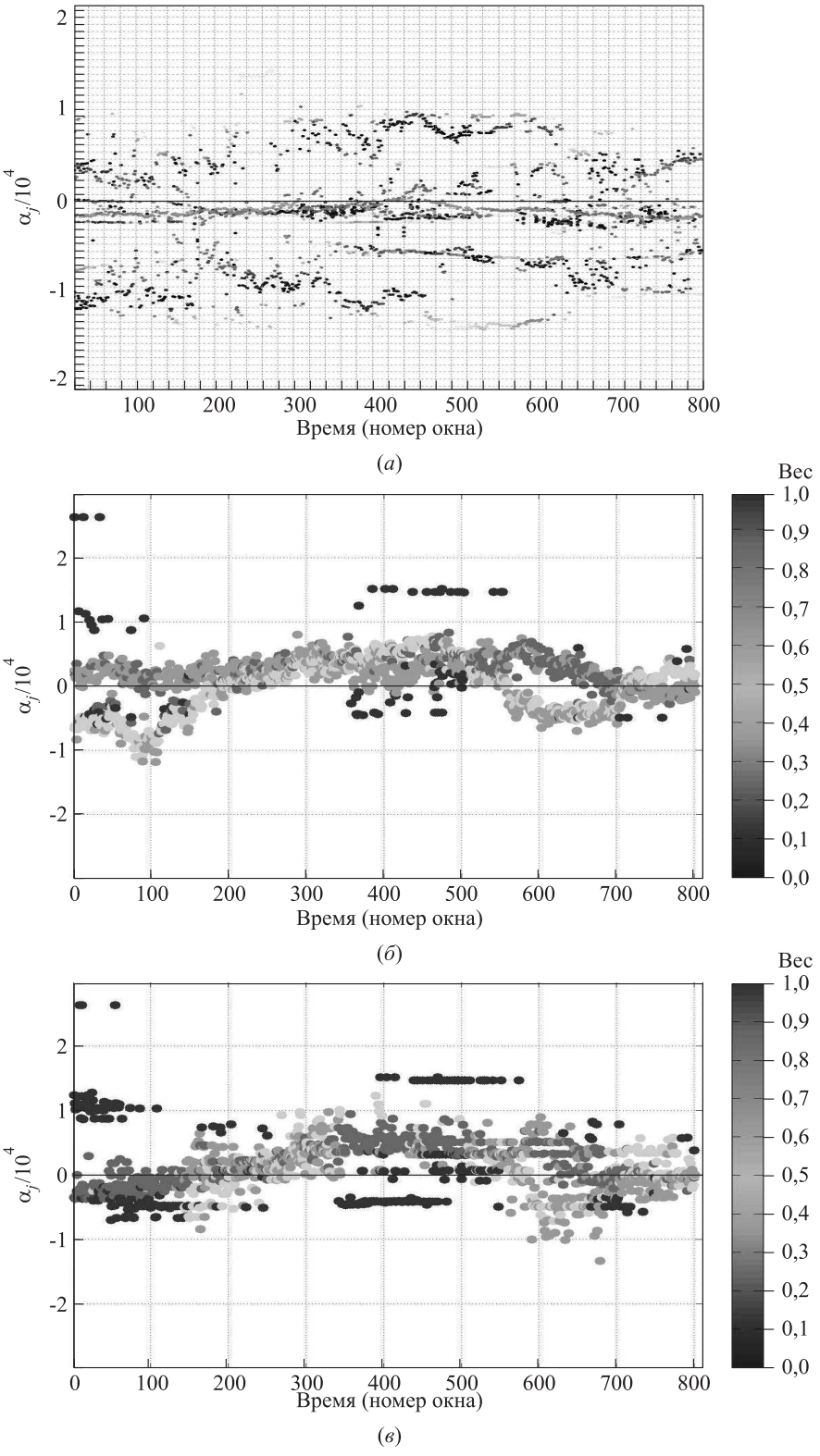


Рис. 20 Портреты динамической волатильности тестовой выборки, полученные: EM-алгоритмом (*а*); SEM-алгоритмом (*б*); медианной модификацией SEM-алгоритма (*в*)

Литература

1. *Королёв В. Ю.* EM-алгоритм, его модификации и их применение к задаче разделения смесей вероятностных распределений. Теоретический обзор. — М.: ИПИРАН, 2007.
2. *Tukey J. W.* A survey of sampling from contaminated distributions // Contributions to probability and statistics. Essays in Honor of Harold Hotelling / Eds. I. Olkin, S. G. Ghurye, W. Hoeffding, W. G. Madow, H. B. Mann.— Stanford: Stanford University Press, 1960. P. 448–485.
3. *Айвазян С. А., Енюков И. С., Мешалкин Л. Д.* Прикладная статистика. Основы моделирования и первичная обработка данных. — М.: Финансы и статистика, 1983.
4. *Королёв В. Ю.* Теория вероятностей и математическая статистика. — М.: Проспект, 2006.
5. *Горшенин А. К., Королёв В. Ю., Турсунбаев А. М.* Медианные модификации EM-алгоритма для разделения смесей вероятностных распределений и их применение к декомпозиции волатильности финансовых индексов // Статистические методы оценивания и проверки гипотез. — Пермь: Изд-во Пермского университета, 2008 (в печати).
6. *Васильев Ф. П.* Методы оптимизации. — М.: Факториал Пресс, 2002.
7. *Колмогоров А. Н.* Метод медианы в теории ошибок // Матем. сборник, 1931. Т. 38, № 3/4. С. 47–50.
8. *Колмогоров А. Н.* Теория вероятностей и математическая статистика. Сб. статей. — М.: Наука, 1986.
9. *Diebolt J., Celeux G.* Asymptotic properties of a stochastic EM algorithm for estimating mixing proportions // Communications in Statistics B: Stochastic Models, 1993. Vol. 9, No. 4. P. 599–613.
10. *Diebolt J., Ip E. H. S.* Stochastic EM: Method and application // Markov chain Monte Carlo in practice / Eds. W. R. Gilks, S. Richardson, D. J. Spiegelhalter. — London: Chapman and Hall, 1996.
11. *Королёв В. Ю.* Статистическая декомпозиция волатильности // Статистические методы оценивания и проверки гипотез. — Пермь: Изд-во Пермского университета, 2007. С. 170–206.
12. *Королёв В. Ю.* Вероятностно-статистический анализ хаотических процессов с помощью смешанных гауссовских моделей. Декомпозиция волатильности финансовых индексов и турбулентной плазмы. — М.: ИПИРАН, 2007.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2008. Т. 2. Вып. 4. С. 48–56

РАСЩЕПЛЕНИЕ СМЕСИ ВЕРОЯТНОСТНЫХ РАСПРЕДЕЛЕНИЙ НА ДВЕ СОСТАВЛЯЮЩИЕ

М. П. Кривенко

Аннотация: Рассматривается представление плотности распределения в виде смеси двух составляющих и задача оценивания параметров этой смеси при имеющихся выборках из самой плотности и из одной из составляющих. Предлагаются два метода построения оценок, строятся соответствующие алгоритмы и проводится их сравнительный анализ.

Ключевые слова: смесь нормальных распределений; расщепление смеси распределений; EM-алгоритм

1 Введение

Рассмотрим представление плотности распределения в виде смеси, состоящей из двух составляющих:

$$g_{B,S}(u) = pg_B(u) + (1-p)g_S(u)\,,\ \text{где}\ p \in [0,\,1]\,, \quad (1)$$

и задачу оценивания элементов этого представления при условии наличия двух выборок: $X_{B,S}$ из $g_{B,S}(u)$ и X_B из $g_B(u)$.

Дадим пример практической ситуации, в рамках которой может возникнуть сформулированная задача [1]. Речь идет об обработке изображений, когда интенсивности серого у пикселов для изображений знаков на некотором фоне отличаются от интенсивностей пикселов самого фона. Необходимо по изображению знаков на некотором фоне (выборка $X_{B,S}$ из $g_{B,S}(u)$) и изображению только фона (выборка X_B из $g_B(u)$) составить представление об уровне фона (оценить p) и описать распределения интенсивностей серого как знаков, так и фона (оценить $g_S(u)$ и $g_B(u)$). После получения перечисленных оценок становится возможным оценить порог T, отделяющий «оптимальным» образом пикселы фона и знаков, и, например, перейти к черно-белому изображению. Важность подобного автоматического метода оценивания порога T с помощью экспериментальных данных для практики объясняется следующим: качество реальных изображений может быть невысоким и существенно меняться в процессе распознавания определенного текста, в силу чего процесс экспертного установления значения T становится достаточно трудоемким, а основанные на черно-белом изображении процедуры распознавания оказываются достаточно эффективными. При этом T понимается как решение уравнения

$$\alpha(T) = \beta(T)\,,$$

где $\alpha(T)$, $\beta(T)$ — ошибки при разделении двух классов, а именно первого класса, соответствующего фону с вероятностью появления p и плотностью распределения $g_B(u)$, и второго класса, соответствующего пикселам знаков с вероятностью появления $1-p$ и плотностью распределения $g_S(u)$.

Рассмотрим (1) как уравнение относительно $g_S(u)$. Если $g_B(u) > 0$ для $\forall u$, то

$$g_S(u) = \frac{g_B(u)}{1-p}\left(\frac{g_{B,S}(u)}{g_B(u)} - p\right)\,.$$

Тогда решение (1) является либо единственным и $p = 0$, либо неединственным и существует для всех $p \in [0, \inf\limits_u(g_{B,S}(u)/g_B(u))]$. Вряд ли такой вывод практически интересен. Введем дополнительные ограничения, позволяющие сначала определиться со значением p, а затем пытаться находить $g_S(u)$. Пусть существуют значения u, для которых $g_S(u) = 0$ и $g_{B,S} \neq 0$. Для таких точек (1) дает уравнение

$$g_{B,S}(u) = pg_B(u)\,, \quad (2)$$

после чего при известных $g_{B,S}(u)$ и $g_B(u)$ становится возможным найти искомое p и, следовательно, $g_S(u)$, если, конечно, соотношения (1) и (2) не становятся противоречивыми для различных точек u.

Приведенные рассуждения носят прикидочный характер, позволяют прояснить идеи, которые положены автором в алгоритмы решения поставленной задачи.

При оценивании элементов представления (1) можно следовать одной из следующих схем:

– от частного к общему, когда сначала по X_B находится оценка $g_B^*(u)$, а затем по $X_{B,S}$ оцениваются величины p^*, $g_S^*(u)$;

[1]Институт проблем информатики Российской академии наук, mkrivenko@ipiran.ru

– от общего к частному, когда сначала по $X_{B,S}$ оценивается $g^*_{B,S}(u)$, а затем с учетом X_B так подбираются p^*, $g^*_B(u)$, $g^*_S(u)$, чтобы

$$p^* g^*_B(u) + (1-p^*) g^*_S(u) \approx g^*_{B,S}(u)\,.$$

Для оценивания $g_{B,S}(u)$, $g_B(u)$ и $g_S(u)$ удобно представить их в виде смеси нормальных распределений с неизвестными параметрами. Таким образом, модель смеси распределений используется как для описания сложной структуры данных, так и для аппроксимации реальных распределений данных с помощью плотностей нормального распределения.

2 Модель смеси нормальных распределений и оценивание ее параметров

Рассмотрим модель смеси распределений, когда плотность некоторого распределения $f(u)$ представима в виде

$$f(u) = \sum_{j=1}^{k} p_{f_j} h\left(u, \vartheta_{f_j}\right)\,. \tag{3}$$

В представлении (3) неизвестными являются все или часть следующих характеристик (параметров смеси): число k элементов смеси (число компонентов смеси), вероятности p_{f_j} элементов смеси (веса элементов смеси), параметры ϑ_{f_j} элементов смеси. Вид плотности $h(u,\vartheta)$ предполагается известным:

$$h(u,\vartheta) \equiv h(u,a,c) = \frac{1}{\sqrt{2\pi}\,c}\,\exp\left\{-\frac{(u-a)^2}{2c^2}\right\}\,.$$

Для оценивания неизвестных параметров (3) используется метод максимального правдоподобия. При решении задачи оценивания параметров смеси далее предполагается, что число k элементов смеси задано.

Согласно принципу максимального правдоподобия центральным становится решение оптимизационной задачи вида

$$\ln L(p_f, \vartheta_f) \equiv \sum_{i=1}^{N} \ln\left(\sum_{j=1}^{k} p_{f_j} h\left(x_i, \vartheta_{f_j}\right)\right) \to \to \max_{\substack{p_{f_1},\ldots,p_{f_k},\ \sum\limits_{j=1}^{k} p_{f_j}=1,\\ \vartheta_{f_1},\ldots,\vartheta_{f_k}}}\,. \tag{4}$$

Наиболее работоспособную общую схему построения процедур, позволяющих находить решения задачи (4), обычно называют EM-алгоритмом [2].

Введем в рассмотрение так называемые апостериорные вероятности q_{ij} принадлежности наблюдения x_i к j-му элементу смеси. Если известны значения параметров p_{f_j} и ϑ_{f_j}, $j=1,\ldots,k$, то при наблюденном значении x_i, $i=1,\ldots,N$, апостериорная вероятность принадлежности этого значения к j-му элементу смеси принимает вид:

$$q_{ij}(x_i, p_{f_j}, \vartheta_{f_j}) = \frac{p_{f_j} h(x_i, \vartheta_{f_j})}{\sum\limits_{l=1}^{k} p_{f_l} h(x_i, \vartheta_{f_l})}\,. \tag{5}$$

Если это не будет искажать смысл, то в записи апостериорных вероятностей будем далее опускать перечисление аргументов, т. е. вместо $q_{ij}(x_i, p_{f_j}, \vartheta_{f_j})$ будем просто записывать q_{ij}. Из определения следует, что для всех допустимых i и j выполняется следующее:

$$q_{ij} \geq 0 \;\text{ и }\; \sum_{j=1}^{k} q_{ij} = 1\,.$$

Преобразуем так логарифм функции правдоподобия, чтобы в получившемся для него выражении фигурировали апостериорные вероятности q_{ij}, а именно при $p_{f_j} h(x_i, \vartheta_{f_j}) \neq 0$ имело место представление:

$$\ln L(p_f, \vartheta_f) = \sum_{j=1}^{k}\sum_{i=1}^{N} q_{ij} \ln p_{f_j} + + \sum_{j=1}^{k}\sum_{i=1}^{N} q_{ij} \ln h(x_i, \vartheta_{f_j}) - \sum_{j=1}^{k}\sum_{i=1}^{N} q_{ij} \ln q_{ij}\,. \tag{6}$$

Для доказательства справедливости этого представления достаточно в правую часть (6) подставить выражение для апостериорной вероятности. Теперь для нахождения оценки параметров смеси распределений появляется возможность сформулировать **EM-алгоритм:**

1. Положить $t=0$ и задать для параметров смеси начальные значения $p^{(0)}_{f_j}$ и $\vartheta^{(0)}_{f_j}$, $j=1,\ldots,k$.
2. По формуле (5) вычислить значения $q^{(t+1)}_{ij}\left(x_i, p^{(t)}_{f_j}, \vartheta^{(t)}_{f_j}\right)$, $i=1,\ldots,N$ и $j=1,\ldots,k$.
3. С помощью выражения (6) определить значения $p^{(t+1)}_{f_j}$ и $\vartheta^{(t+1)}_{f_j}$ для $j=1,\ldots,k$ из условия максимизации отдельно каждого из первых двух слагаемых правой части (6), поскольку при фиксированных апостериорных вероятностях первое слагаемое $\sum\limits_{j=1}^{k}\sum\limits_{i=1}^{N} q_{ij} \ln p_{f_j}$ зависит только от параметров p_f, а второе слагаемое $\sum\limits_{j=1}^{k}\sum\limits_{i=1}^{N} q_{ij} \ln h(x_i, \vartheta_{f_j})$ зависит только от параметров ϑ_f.

4. Завершить процесс нахождения оценки параметров смеси, если эти оценки являются подходящими, или положить $t = t + 1$ и перейти к шагу 2 в противном случае.

Решение оптимизационной задачи

$$\sum_{j=1}^{k}\sum_{i=1}^{N} q_{ij} \ln p_{f_j} \to \max_{\substack{p_{f_1},\ldots,p_{f_k} \\ \sum_{j=1}^{k} p_{f_j}=1}}$$

имеет вид:

$$p^*_{f_j} = \frac{w_j}{N}\,, \quad j = 1,\ldots,k\,, \text{ где } w_j = \sum_{i=1}^{N} q_{ij}\,.$$

Для доказательства этого достаточно рассмотреть функцию Лагранжа.

Решение оптимизационной задачи

$$\sum_{j=1}^{k}\sum_{i=1}^{N} q_{ij} \ln h(x_i, \vartheta_{f_j}) \to \max_{\vartheta_{f_j},\ldots,\vartheta_{f_k}}$$

распадается на решения для $j = 1,\ldots,k$ отдельных задач:

$$\sum_{i=1}^{N} q_{ij} \ln h(x_i, \vartheta_{f_j}) \to \max_{\vartheta_{f_j}}, \tag{7}$$

в случае нормального распределения они могут быть получены аналитическим путем:

$$a^*_{f_j} = \frac{\sum\limits_{i=1}^{N} q_{ij} x_i}{w_j}\,, \quad \left(c^2_{f_j}\right)^* = \frac{\sum\limits_{i=1}^{N} q_{ij}(x_i - a^*_{f_j})^2}{w_j}\,.$$

Постановка задачи в виде (7) без дополнительных ограничений на параметр ϑ_{f_j} может привести к появлению бесконечно большого вклада в значение функции правдоподобия. Причина этого может состоять в том, что к j-му элементу смеси в ходе работы EM-алгоритма могут быть отнесены либо единственное наблюденное значение, либо совпадающие наблюденные значения, при этом выборочная дисперсия оказывается равной нулю. В этом случае можно ограничивать множество допустимых значений дисперсии и тем самым ограничивать значение максимума функции правдоподобия. Таким образом, задача (7) принимает вид:

$$\sum_{i=1}^{N} q_{ij} \ln h(x_i, a_{f_j}, c_{f_j}) \to \max_{\substack{a_{f_j}, c_{f_j}, \\ c_{f_j} \ge c_{\min}}}\,,$$

где $c_{\min}$ задано, а ее решение, согласно [1], имеет следующий вид:

$$a^{**}_{f_j} = a^*_{f_j}\,, \quad \left(c^2_{f_j}\right)^{**} = \begin{cases} \left(c^2_{f_j}\right)^*\,, & \left(c^2_{f_j}\right)^* > c^2_{\min}\,, \\ c^2_{\min}\,, & \left(c^2_{f_j}\right)^* \le c^2_{\min}\,. \end{cases}$$

Заметим, что если вместо (4) решается задача

$$\sum_{i=1}^{N} \ln\left(\sum_{j=1}^{k} p_{f_j} h\left(x_i, \vartheta_{f_j}\right)\right) \to \max_{\substack{p_{f_1},\ldots,p_{f_k}, \\ \sum_{j=1}^{k} p_{f_j}=1}}\,,$$

то это просто приводит к упрощению 3-го шага EM-алгоритма.

3 Процедуры расщепления смеси

Реализация описанной схемы от частного к общему оказалась неработоспособной, так как при нахождении p^*, $g^*_S(u)$ при уже полученной $g^*_B(u)$ часто возникает следующая ситуация: p^* устремляется к нулю и $g^*_S(u)$ становится оценкой для $g_{B,S}(u)$, а не для $g_S(u)$.

При оценивании элементов модели (1) в соответствии со схемой от общего к частному сначала для выборки $X_{B,S}$ решается задача нахождения оценки $g^*_{B,S}(u)$ в виде смеси

$$g^*_{B,S}(u) = \sum_{j=1}^{k_{B,S}} p^*_{B,S_j} h\left(u, \vartheta^*_{B,S_j}\right)\,, \tag{8}$$

а затем решается задача расщепления смеси $g^*_{B,S}(u)$ на две части вида (1). Последнее предлагается реализовать одним из двух возможных способов:

(1) одновременное оценивание p^*, $g^*_B(u)$, $g^*_S(u)$ путем выбора такого набора индексов элементов смеси (8), при котором достигается максимум функции правдоподобия для выборки X_B;

(2) последовательное оценивание сначала значения p^*, затем вероятностей $p^*_{B_j}$ в разложении

$$g^*_B(u) = \sum_{j=1}^{k_{B,S}} p^*_{B_j} h\left(u, \vartheta^*_{B,S_j}\right)$$

для выборки X_B, а затем формирование оценок вероятностей $p^*_{S_j}$ в разложении

$$g^*_S(u) = \sum_{j=1}^{k_{B,S}} p^*_{S_j} h\left(u, \vartheta^*_{B,S_j}\right)$$

с помощью представления

$$g^*_S(u) \approx \frac{1}{1-p^*}\left(g^*_{B,S}(u) - p^* g^*_B(u)\right)\,.$$

При реализации 1-го способа можно предложить два варианта выбора набора индексов элементов смеси (8):

- полный перебор всех возможных комбинаций индексов и выбор из них той комбинации, для которой достигается наибольшее значение функции правдоподобия (оптимальный выбор);
- последовательное построение комбинации путем перемещения в формируемую смесь $g_B^*(u)$ из оставшихся элементов смеси (8) того элемента, для которого сформированная комбинация приводит к наибольшему значению функции правдоподобия; завершение этого процесса происходит, когда значение функции правдоподобия станет уменьшаться (субоптимальный выбор).

В данной работе рассматривается использование только оптимального выбора набора индексов.

Дадим более подробное описание конкретных вариантов соответствующих процедур обработки данных. Их работа определяется следующими основными входными параметрами:

- заданная выборка $X_{B,S}$;
- заданная выборка X_B;
- заданные параметры EM-алгоритма (число элементов смеси k, критическое значение дисперсии $c^2_{\min}$, максимальное число итераций, ошибка нахождения максимума функции правдоподобия);

а также выходными параметрами:

- формируемая оценка p^*;
- формируемые оценки параметров смеси g_B^*;
- формируемые оценки параметров смеси g_S^*.

3.1 Процедура одновременного оценивания

В основе этого алгоритма лежит предположение, что составляющие $g_S(u)$ и $g_B(u)$ «отделимы» друг от друга в том смысле, что в разложении $g_{B,S}(u)$ по некоторой системе плотностей одна его часть относится к $g_S(u)$, а другая — к $g_B(u)$. Поэтому группирование элементов смеси с целью одновременного формирования составляющих $g_S^*(u)$ и $g_B^*(u)$ может быть реализовано с помощью следующей последовательности действий (**алгоритм А1**):

1. Оценивание с помощью EM-алгоритма параметров смеси (8) по выборке $X_{B,S}$.
2. Выбор подмножества множества элементов $\left\{p^*_{B,S_j} h\left(u, \vartheta^*_{B,S_j}\right), j = 1, \ldots, k_{B,S}\right\}$ так, чтобы при X_B получить максимальное значение функции правдоподобия (см. далее (9)).
3. Нахождение значения p^* и $g_B^*(u)$ (см. далее (10)).
4. Формирование оценки для $g_S^*(u)$ (см. далее (11)).

Пусть I_B — подмножество множества индексов элементов смеси заданного набора $\{p^*_{B,S_j} h(u, \vartheta^*_{B,S_j}), j = 1, \ldots, k_{B,S}\}$, а N — объем выборки X_B. Обозначив с целью упрощения записи

$$p_j = p^*_{B,S_j}\,,\ a_j = a^*_{B,S_j}\,,\ c_j = c^*_{B,S_j}\,,\ j = 1, \ldots, k_{B,S}\,,$$

выпишем формулы для подсчета функции правдоподобия:

$$\begin{aligned}
&\ln L(p, a, c|I_B) = \\
&\quad = \sum_{i=1}^{N} \ln \left(\sum_{j \in I_B} \frac{p_j}{\sum\limits_{j \in I_B} p_j} h(x_i, a_j, c_j) \right) = \\
&= \sum_{i=1}^{N} \ln \left(\sum_{j \in I_B} p_j \frac{1}{\sqrt{2\pi}\, c_j} \exp\left[-\frac{(x_i - a_j)^2}{2c_j^2} \right] \right) - \\
&\quad - N \ln \sum_{j \in I_B} p_j = \sum_{i=1}^{N} \ln \left(\sum_{j \in I_B} h_{ij} \right) - \\
&\quad - N \ln \sum_{j \in I_B} p_j - \frac{N}{2} \ln(2\pi)\,, \qquad (9)
\end{aligned}$$

где

$$h_{ij} = \frac{p_j}{c_j} \exp\left[-\frac{(x_i - a_j)^2}{2c_j^2} \right]$$

для $i = 1, \ldots, N$ и $j = 1, \ldots, k_{B,S}$.

После того как найдено подмножество $\{p^*_{B_j} h(u, \vartheta^*_{B_j}),\ j = 1, \ldots, k_B\}$ и соответствующее I_B^*, оценки для $g_B^*(u)$ и p^* принимают следующий вид:

$$\begin{aligned}
p^* &= \sum_{j \in I_B^*} p^*_{B,S_j} \equiv \sum_{j=1,\ldots,k_B} p^*_{B_j}\,; \\
g_B^*(u) &= \frac{1}{p^*} \sum_{j \in I_B^*} p^*_{B,S_j} h\left(u, \vartheta^*_{B,S_j}\right)\,.
\end{aligned} \qquad (10)$$

И, наконец,

$$\begin{aligned}
g_S^*(u) = &\left\{ p^*_{B,S_j} h\left(u, \vartheta^*_{B,S_j}\right)\,,\ j = 1, \ldots, k_{B,S} \right\} \setminus \\
&\left\{ p^*_{B_j} h\left(u, \vartheta^*_{B_j}\right)\,,\ j = 1, \ldots, k_B \right\}\,. \qquad (11)
\end{aligned}$$

3.2 Процедура последовательного оценивания

В основе этого алгоритма лежит предположение, что существует множество значений u, для которых $g_S(u) = 0$ и $g_{B,S}(u) \neq 0$. В данной работе это множество — область $u \in [b, +\infty)$, где b — некоторое значение, и задается оно либо априори (например, исходя из сущности решаемой задачи), либо по имеющимся данным (например, как найденная по выборке X_B оценка среднего распределения $g_B(u)$ или как найденная по выборке X_B оценка квантиля распределения $g_B(u)$ некоторого порядка). Выбирая тем или иным способом b, необходимо быть только уверенным, что $g_S(u) \approx 0$ при $u > b$.

Тогда нахождение требуемых оценок осуществляется с помощью следующих шагов (**алгоритм А2**):

1. Нахождение значения p^* (см. далее (12)).
2. Оценивание с помощью EM-алгоритма параметров смеси (8) по выборке $X_{B,S}$.
3. Оценивание с помощью EM-алгоритма параметров смеси для $g_B^*(u)$ по выборке X_B путем коррекции только весов элементов смеси, найденной на 2-м шаге.
4. Оценивание параметров смеси для $g_B^*(u)$ путем «вычитания» из оценки $g_{B,S}^*$ взвешенной оценки $p^* g_B^*(u)$.

Пусть $1 - \overline{G}_{B,S}(u)$ — функция распределения для $g_{B,S}(u)$, а $1 - \overline{G}_B(u)$ — для $g_B(u)$. Тогда имеем следующие соотношения:

$$\overline{G}_{B,S}(b) = \int\limits_b^{+\infty} g_{B,S}(u)\,du = \int\limits_b^{+\infty} ((1-p)\,g_S(u) + \\ + pg_B(u))\,du \approx \int\limits_b^{+\infty} pg_B(u)\,du = p\overline{G}_B(b)\,.$$

Отсюда получаем:

$$\widetilde{p}^* = \frac{\overline{G}_{B,S}^*(b)}{\overline{G}_B^*(b)}\,; \\ p^* = \begin{cases} \widetilde{p}^*\,, & \overline{G}_{B,S}^*(b) \neq 0 \text{ и } \widetilde{p}^* \leq 1\,, \\ 1\,, & \text{в иных случаях}\,, \end{cases} \tag{12}$$

где оценка $\overline{G}_{B,S}^*(b)$ получена по выборке $X_{B,S}$, а $\overline{G}_B^*(b)$ — по выборке X_B.

После выполнения 2-го и 3-го шагов алгоритма А2 получены следующие два разложения:

$$g_{B,S}^*(u) = \sum_{j=1}^{k_{B,S}} p_{B,S_j}^* h\left(u, \vartheta_{B,S_j}^*\right)\,;$$

$$g_B^*(u) = \sum_{j=1}^{k_{B,S}} p_{B_j}^* h\left(u, \vartheta_{B,S_j}^*\right)\,.$$

Тогда оценка для составляющей $g_S(u)$ принимает вид:

$$g_S^*(u) = \sum_{j=1}^{k_{B,S}} p_{S_j}^* h(u, \vartheta_{B,S_j})\,,$$

где для нахождения $p_{S_j}^*$ необходимо последовательно получить следующее:

$$\widetilde{p}_{S_j}^* = \begin{cases} p_{B,S_j}^* - p^* p_{B_j}^*\,, & p_{B,S_j}^* - p^* p_{B_j}^* > 0\,, \\ 0\,, & p_{B,S_j}^* - p^* p_{B_j}^* \leq 0\,, \end{cases} \\ \text{для } j = 1, \ldots, k_{B,S}\,;$$

$$\widetilde{P}_S^* = \sum_{j=1}^{k_{B,S}} \widetilde{p}_{S_j}^*\,;$$

$$p_{S_j}^* = \frac{\widetilde{p}_{S_j}^*}{\widetilde{P}_S^*} \text{ для } j = 1, \ldots, k_{B,S}\,.$$

Мерой точности такого приближенного решения (1) как уравнения относительно $g_S(u)$ может служить следующая величина:

$$\varepsilon = (1 - p^*) \sum_{\substack{j=1,\ldots,k_{B,S} \\ p_{B,S_j}^* - p^* p_{B_j}^* < 0}} \left| p_{B,S_j}^* - p^* p_{B_j}^* \right|\,.$$

Для оценки p^* можно, в принципе, построить ее распределение. Действительно, оценки $\overline{G}_B^*(b)$ и $\overline{G}_{B,S}^*(b)$ основываются на числе успехов в последовательности испытаний Бернулли: для $\overline{G}_B^*(b)$ это $\widetilde{v}$ в N испытаниях с вероятностью успеха q, для $\overline{G}_{B,S}^*(b)$ это $\widetilde{w}$ в M испытаниях с вероятностью успеха pq. Здесь N — объем выборки X_B и $q = \int\limits_b^{+\infty} g_B(u)\,du$, M — объем выборки $X_{B,S}$ и $pq = \int\limits_b^{+\infty} g_{B,S}(u)\,du$.

Предполагается, что выборки X_B и $X_{B,S}$ являются независимыми. Тогда совместное распределение числа успехов в двух испытаниях Бернулли есть

$$\Pr\{\widetilde{V} = \widetilde{v}, \widetilde{W} = \widetilde{w}\} = \\ = \binom{N}{\widetilde{v}} q^{\widetilde{v}} (1-q)^{N-\widetilde{v}} \binom{M}{\widetilde{w}} (pq)^{\widetilde{w}} (1-pq)^{M-\widetilde{w}}\,.$$

Это позволяет в явном виде записать вероятность появления любого допустимого значения для W/V,

где $W = \widetilde{W}/M$ и $v = \widetilde{V}/N$, а следовательно, получить все необходимые характеристики случайной величины p^*, определенной в (12). Асимптотический подход может дать более простые результаты. Если через $\Phi(a, c^2)$ обозначить нормальное распределение с параметрами (a, c^2), а знак $\Leftarrow$ употреблять между обозначениями случайной величины и ее распределением, то при $M, N \to \infty$ имеем:

$$V \Leftarrow \Phi\left(q, \frac{q(1-q)}{N}\right);$$
$$W \Leftarrow \Phi\left(pq, \frac{pq(1-pq)}{M}\right).$$

Найдем вероятность $\Pr\{W - uV \le y\}$ для некоторых значений u, y. Имеем:

$$W - uV \Leftarrow \Phi\left(pq - uq, \frac{pq(1-pq)}{M} + u^2\frac{q(1-q)}{N}\right)$$

и, следовательно,

$$\Pr\{W - uV \le y\} = \Phi_{0,1}\left(\frac{y - (pq - uq)}{\sqrt{pq(1-pq)/M + u^2 q(1-q)/N}}\right),$$

где $\Phi_{0,1}$ — функция стандартного нормального распределения. Но

$$\Pr\{W - uV \le 0\} = \Pr\{W \le uV\} \approx \Pr\left\{\frac{W}{V} \le u\right\}.$$

Таким образом,

$$\Pr\{P^* \le u\} \equiv \Pr\left\{\frac{W}{v} \le u\right\} \approx$$
$$\approx \Phi_{0,1}\left(\frac{uq - pq}{\sqrt{pq(1-pq)/M + u^2 q(1-q)/N}}\right). \quad (13)$$

С помощью (13) можно, в частности, проанализировать зависимость свойств оценки p^* от выбора значения q. В качестве выборочных свойств оценки рассмотрим смещение и стандартное отклонение, а именно: если p^* есть оценка параметра p, принимающего значение p_0, то смещение этой оценки есть $\mathrm{Bias}(p^*) = E\{p^*\} - p_0$; а стандартное отклонение — $\mathrm{St}D(p^*) = \sqrt{E\{(p^* - E\{p^*\})^2\}}$. Для вычисления смещения и стандартной среднеквадратичной ошибки используем (13) и приближенный метод вычисления соответствующих интегралов. На рис. 1 дан график зависимостей введенных выборочных характеристик оценки от значения q для $p_0 = 50\%$, $M = N = 800$. Из него, в частности, видно, что с точки зрения качества оценки p^* значение q следует выбирать как можно большим; но при этом не следует забывать, что с точки зрения корректности отделения $g_S(u)$ и $g_B(u)$ значение q следует выбирать как можно меньшим (соответствующее значение b должно быть как можно большим).

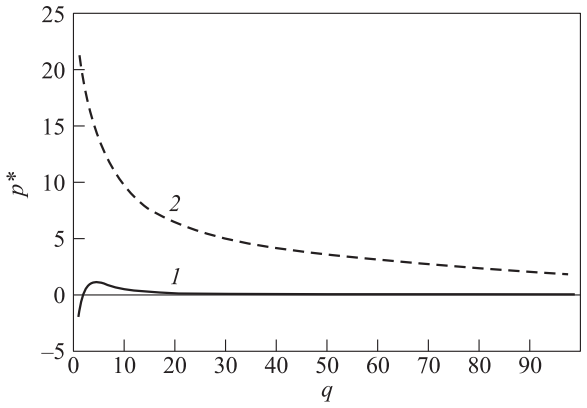


Рис. 1 Зависимости свойств оценки p^* от выбора q: *1* — $\mathrm{Bias}(p^*)$; *2* — $\mathrm{St}D(p^*)$

4 Эксперименты

Цель экспериментов состояла в сравнительном анализе свойств построенных алгоритмов. В качестве модели обрабатываемых данных была выбрана смесь распределений $g_B(u)$ и $g_S(u)$. Каждая составляющая $g_B(u)$ и $g_S(u)$ описывалась, в свою очередь, с помощью смесей нормальных распределений следующего общего вида:

$$g_B(u) = \sum_{j=1}^{16} p_{B_j} h\left(u, a_{B_j}, c_{B_j}\right);$$
$$g_S(u) = \sum_{j=1}^{6} p_{S_j} h\left(u, a_{S_j}, c_{S_j}\right)$$

(значения параметров приведены в табл. 1); с помощью этих смесей аппроксимировались распределения реальных данных, встречающихся при распознавании зашумленного изображения текста.

Качество процедур расщепления смеси на две составляющие характеризовалось выборочными свойствами оценок веса p^* и порога T^*, а также временем обработки выборок. Параметрами задачи сравнительного анализа свойств алгоритмов являлись следующие: значение p_0, значение объема выборок (объемы $X_{B,S}$ и X_B здесь и далее брались равными), значение k — числа элементов смеси, принятого для описания выборочных распределений. Для обеих процедур многократно проводились эксперименты по моделированию выборок из $g_{B,S}(u) = (1 - p_0)g_S(u) + p_0 g_B(u)$ и построению оценок p^*(A1) и p^*(A2). После получения

Таблица 1 Параметры моделей

j	$g_B(u)$			$g_S(u)$		
	p_{B_j}	a_{B_j}	$c^2_{B_j}$	p_{S_j}	a_{S_j}	$c^2_{S_j}$
1	0,01	120,280	45,085	0,15	77,000	0,050
2	0,02	129,915	26,361	0,20	86,263	28,335
3	0,02	135,846	5,493	0,35	103,902	70,201
4	0,08	143,709	6,024	0,15	120,280	45,085
5	0,11	150,824	2,660	0,10	129,915	26,361
6	0,04	153,636	0,239	0,05	135,846	5,493
7	0,08	156,556	0,264			
8	0,05	158,000	0,050			
9	0,07	159,000	0,050			
10	0,08	160,000	0,050			
11	0,08	161,000	0,050			
12	0,10	162,000	0,050			
13	0,08	164,000	0,050			
14	0,05	165,000	0,050			
15	0,05	166,000	0,050			
16	0,08	167,418	0,482			

100 пар этих значений по ним строились оценки для $\mathrm{Bias}(p^*(\mathrm{A1}))$ и $\mathrm{Bias}(p^*(\mathrm{A2}))$, $\mathrm{St}D(p^*(\mathrm{A1}))$ и $\mathrm{St}D(p^*(\mathrm{A2}))$ (с целью упрощения записи далее под $\mathrm{Bias}(\ldots)$ и $\mathrm{St}D(\ldots)$ понимаются их оценки). Далее в плоскости значений $(\mathrm{abs}(\mathrm{Bias}(p^*(\mathrm{A2})))-$ $-\ \mathrm{abs}(\mathrm{Bias}(p^*(\mathrm{A1}))),\ \mathrm{St}D(p^*(\mathrm{A2}))\ -\ \mathrm{St}D(p^*(\mathrm{A1})))$ для $p_0 = 25\%, 50\%, 75\%$, $N = M = 200, 800, 3\,200, 12\,800$, $k = 8, 12, 16$ отмечались точки, соответствующие результатам применения двух процедур.

Аналогичные действия проводились для оценок T^*; полученные результаты представлены в графическом виде на рис. 2. Из них на множестве рассмотренных вариантов видно преимущество последовательного оценивания (процедура A2) по сравнению с процедурой одновременного оценивания (процедура A1) параметров расщепленной смеси, так как показатели качества оценок в основном выше для A2 по сравнению с A1.

Кроме того, временные показатели A2 также лучше, чем у A1, что отображено на рис. 3, где в плоскости (log(среднее время A1), log(среднее время A2)) точками отмечены перечисленные варианты моделирования и обработки данных. Таким образом, предпочтение следует отдать последовательной процедуре оценивания параметров расщепленной смеси.

Самостоятельный интерес представляет поведение качества оценок при увеличении объема выборок, соответствующие результаты представлены на рис. 4 и 5.

Обращает на себя внимание «необычное» поведение свойств оценок при различных значени-

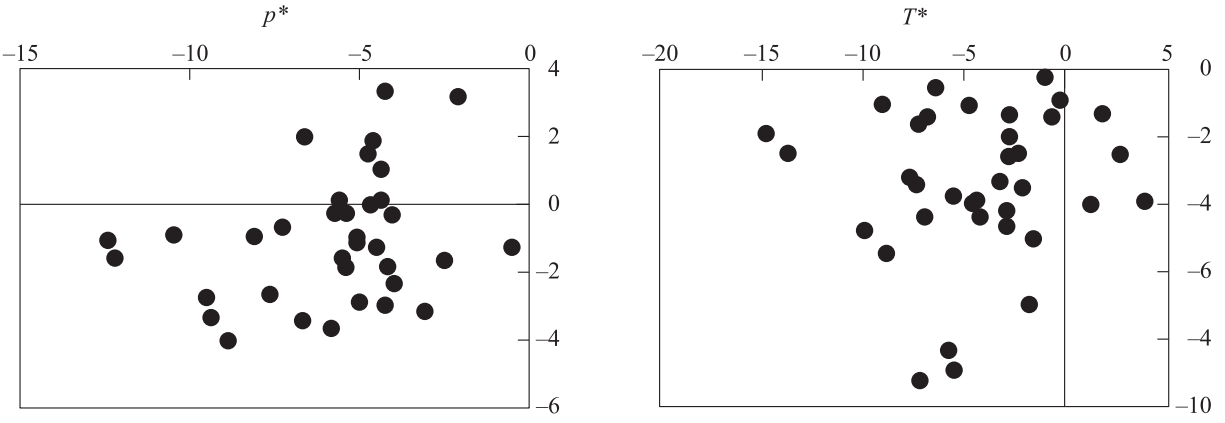


Рис. 2 Сравнительные характеристики качества оценок p^* и T^*

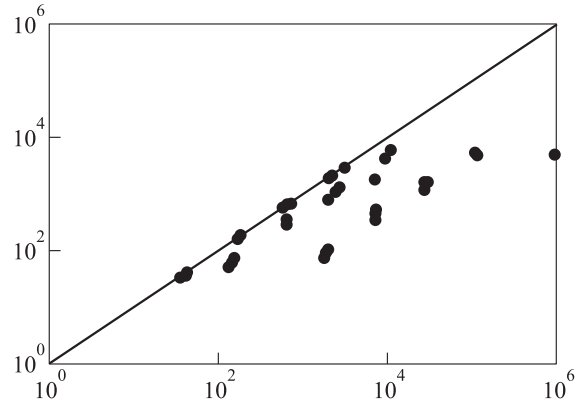


Рис. 3 Сравнительные характеристики среднего времени расщепления смеси

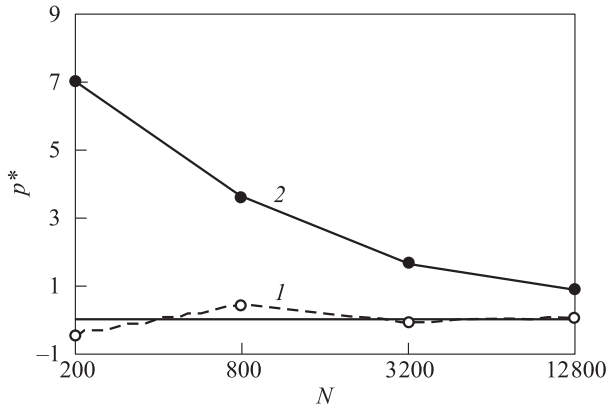
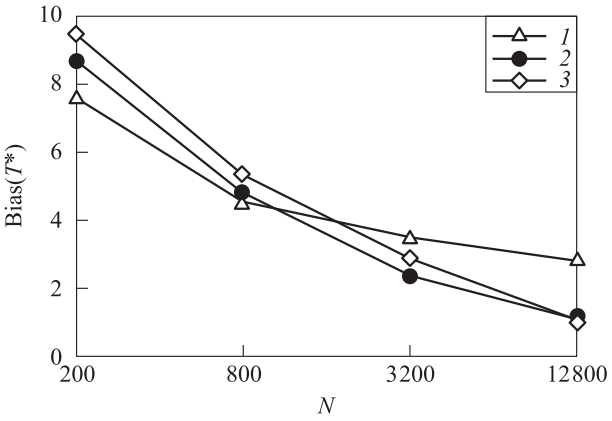


Рис. 4 Зависимости свойств оценки p^* от объема выборок N: *1* — $\mathrm{Bias}(p^*)$; *2* — $\mathrm{St}D(p^*)$



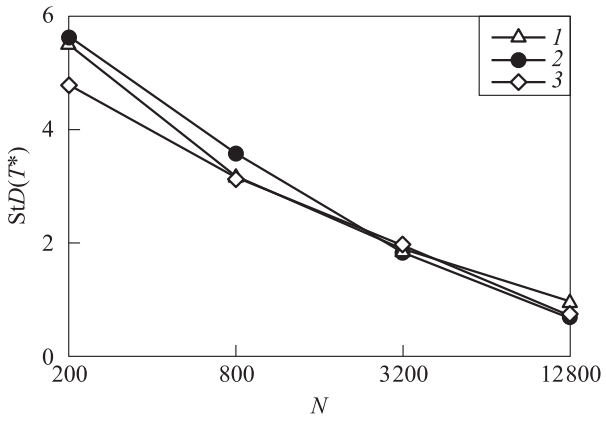


Рис. 5 Зависимости свойств оценки T^* от объема выборок N: *1* — $k = 8$; *2* — $k = 12$; *3* — $k = 16$

ях числа элементов смеси k. Представляется, что увеличение этого параметра должно приводить к улучшению аппроксимации и поэтому к более качественным результатам (заметим, что оценка p^* не зависит от k). В связи с чем отдельно проводилось моделирование зависимости свойств T^* от числа элементов смеси (рис. 6). Полученные результаты позволяют сформулировать следующее предположение: существует некоторое оптимальное значение k, позволяющее наилучшим образом решить задачу аппроксимации распределений $g_B(u)$ и $g_S(u)$. При этом, к счастью, для практических применений увеличение значения k несущественно сказывается на свойствах получающихся оценок.

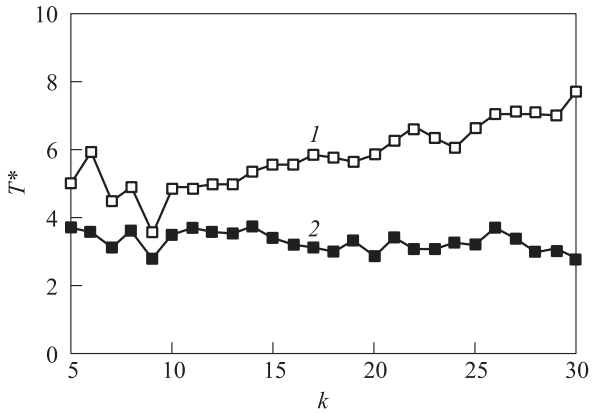


Рис. 6 Зависимости свойств оценки T^* от числа элементов смеси k: *1* — $\mathrm{Bias}(T^*)$; *2* — $\mathrm{St}D(T^*)$

5 Заключение

Полученные решения задачи расщепления смеси на две составляющие представляют самостоятельный интерес, достаточны для их применения на практике. Проведенные эксперименты не только продемонстрировали работоспособность

предложенных решений, но и дали схемы исследования в духе бутстреп-метода свойств оценок, существенно зависящих от множества таких параметров, как объемы выборок, значения q (или b, где $q = \int\limits_b^{+\infty} g_B(u)\,du$), параметры EM-алгоритма (число элементов смеси k, критическое значение дисперсии $c^2_{\min}$, максимальное число итераций, ошибка нахождения максимума функции правдоподобия).

Литература

1. *Кривенко М. П.* Распознавание элементов изображения, имеющих различные размеры // Системы и средства информатики. — М.: ИПИ РАН, 2007. Вып. 17. С. 30–51.
2. *McLachlan G. J., Peel D.* Finite mixture models. — New York: Wiley, 2000. 419 p.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2008. Т. 2. Вып. 4. С. 57–65

НЕОДНОРОДНЫЕ РЕКУРРЕНТНЫЕ МОДЕЛИ ИЗМЕНЕНИЯ НАДЕЖНОСТИ МОДИФИЦИРУЕМЫХ СИСТЕМ. НЕПРЕРЫВНОЕ ВРЕМЯ*

С. В. Артюхов[1], В. Ю. Королёв[2]

Аннотация: В статье с помощью оценок скорости сходимости в предельных теоремах для обобщенных дважды стохастических пуассоновских процессов (обобщенных процессов Кокса) уточняется асимптотическое поведение надежности сложных модифицируемых технических и информационных систем в рамках неоднородных рекуррентных моделей изменения надежности с непрерывным временем.

Ключевые слова: обобщенный дважды стохастический пуассоновский процесс; обобщенный процесс Кокса; модель изменения надежности; коэффициент готовности; гарантированные доверительные границы

1 Введение

Одним из факторов, оказывающих существенное воздействие на качество функционирования технической или информационной системы (ТИС), является модификация отдельных компонентов (узлов) системы. Такая модификация может быть обусловлена как необходимостью замены вышедшего из строя узла, так и желанием улучшить характеристики функционирования системы за счет замены узла устаревшего образца более современным (более совершенным или надежным). Учет этого фактора при анализе надежности сложных ТИС очень важен, поскольку во многих случаях после ремонта или замены отказавшего элемента ТИС необходимо отказаться от предположения о том, что после ремонта вся система останется такой же, как и до ремонта. Только в этом случае можно получить оценки надежности ТИС, приемлемые по точности для практических нужд службы технической эксплуатации ТИС. Однако при отказе от предположения об идентичности системы до и после ее модернизации становятся неприемлемыми традиционные методы анализа характеристик надежности восстанавливаемых систем (см. ГОСТ 27.002–89). Поэтому целью данной работы является описание метода построения оценок показателей надежности ТИС с помощью математических моделей, учитывающих возможное изменение характеристик ТИС после каждой модификации.

При изучении надежностных характеристик модифицируемых систем возможны как минимум два подхода. Первый из них заключается в том, что параметры распределения времени безотказной работы считаются функциями номера модификации (отказа, восстановления), что приводит к моделям с дискретным временем (дискретным моделям изменения надежности). В рамках второго подхода параметры распределения времени безотказной работы считаются функциями астрономического времени, что приводит к моделям с непрерывным временем (непрерывным моделям изменения надежности).

Результаты данной работы дополняют основы математической теории роста надежности модифицируемых систем, изложенные в [1–3]. В последней из упомянутых книг был введен класс так называемых *рекуррентных* моделей изменения надежности модифицируемых систем (также см. гл. 2 в [4]) и изучены некоторые аналитические свойства этих моделей. В настоящей работе эти результаты будут обобщены и уточнены.

При анализе надежности ТИС довольно часто вместо такой характеристики надежности, как время безотказной работы системы, удобно иметь дело непосредственно с интегральным параметром, интерпретируемым как текущая надежность системы. В качестве такого интегрального показателя надеж-

*Работа выполнена при поддержке Российского фонда фундаментальных исследований, гранты 08–01–00345, 08–01–00363, 08–07–00152.

[1]ООО «Внешпромбанк», ArtyuhovSV@yandex.ru

[2]Московский государственный университет, факультет ВМиК; Институт проблем информатики Российской академии наук, vkorolev@comtv.ru

ности ТИС в данной работе будет рассматриваться нестационарный аналог коэффициента готовности. Рассмотрим произвольную систему, на вход которой подаются некоторые сигналы (например, команды оператора или внешние воздействия). Реакция системы на поданные сигналы может быть либо правильной (корректной), либо неправильной (некорректной). В каждый момент времени t надежность системы можно характеризовать параметром $P(t)$ — вероятностью того, что на сигнал, поданный на вход системы в момент t, система отреагирует правильно. По смыслу такая характеристика надежности ближе всего к традиционно используемому *коэффициенту готовности*. В случайные моменты времени $Y_0 = 0 \le Y_1 \le Y_2 \le \ldots$ система подвергается (мгновенной) модификации, в результате чего изменяется параметр $P(t)$. Предположим, что траектории процесса $P(t)$ непрерывны справа и кусочно-постоянны, так что $P(t) = P(Y_j)$ при $t \in [Y_j,\, Y_{j+1})$, $j \ge 1$.

Задача прогнозирования поведения процесса $P(t)$ чрезвычайно важна. Описанная выше очень общая схема может быть переформулирована в терминах, традиционных для столь разных областей знания, как медицина, программирование или менеджмент. Например, в программировании параметр $P(t)$ можно рассматривать как надежность программного обеспечения, в которое по ходу отладки в моменты $Y_0 = 0 \le Y_1 \le Y_2 \le \ldots$ вносятся изменения для исправления замеченных ошибок. Оценивание $P(t)$ и прогнозирование поведения этого параметра здесь важно как для оценивания надежности всего комплекса, составной частью которого является программное обеспечение, так и для прогнозирования продолжительности отладки (более подробно об этом см. в книгах [1–3]). В медицине параметр $1 - P(t)$ (называемый индексом летальности) характеризует вероятность летального исхода в момент времени t для пациента, организм которого в моменты $Y_0 = 0 \le Y_1 \le Y_2 \le \ldots$ подвергается какому-либо медицинскому вмешательству (операции, инъекции, приему лекарств и т. п.). Здесь прогнозирование $P(t)$ чрезвычайно важно с точки зрения принятия решений о стратегии лечения. Наконец, в менеджменте параметр $P(t)$ может характеризовать надежность и дееспособность коллектива, организации или предприятия, структура которых в моменты времени $Y_0 = 0 \le Y_1 \le Y_2 \le \ldots$ претерпевает изменения. Будем предполагать, что в результате каждой модификации системы параметр $P(t)$ изменяется случайным (непредсказуемым) образом.

В данной статье будет рассмотрена зависимость надежности модифицируемой системы не от номера модификации, а от реального времени. Такие задачи представляют собой, пожалуй, больший практический интерес, нежели задачи с дискретным временем. Однако в таком случае необходимо сделать некоторые предположения о распределении случайных моментов времени $\{Y_0 = 0, Y_1, Y_2, \ldots\}$, в которые осуществляются модификации системы. Для начала предположим, что последовательность $\{Y_0 = 0, Y_1, Y_2, \ldots\}$ представляет собой однородный пуассоновский точечный процесс с некоторой интенсивностью $\lambda > 0$. Такое предположение типично, если считать, что моменты модификаций абсолютно хаотично рассредоточены на временной оси (хорошо известно, что совместное условное распределение точек пуассоновского потока на некотором интервале $[a,\, b]$ при условии, что на этом интервале осуществилось ровно n событий потока, совпадает с распределением вариационного ряда, построенного по выборке объема n из равномерного на $[a,\, b]$ распределения). Более того, случайные длины интервалов времени между точками скачков пуассоновского процесса независимы и имеют одно и то же экспоненциальное распределение, которое, как известно, обладает максимальной дифференциальной энтропией среди всех распределений, сосредоточенных на неотрицательной полуоси и имеющих конечное математическое ожидание. Дифференциальная же энтропия является вполне адекватной мерой неопределенности. Таким образом, пуассоновский процесс в наибольшей мере соответствует общепринятым представлениям о наиболее непредсказуемом, хаотическом размещении точек на вещественной прямой. Однако однородный пуассоновский процесс характеризуется постоянной интенсивностью — средним количеством случайных точек, попавших в интервал времени единичной длины.

Однако на практике существенно чаще встречаются ситуации, в которых интенсивность процесса модификации системы непостоянна.

Предположим, что точки $\{Y_0 = 0, Y_1, Y_2, \ldots\}$, в которые осуществляются модификации системы, образуют дважды стохастический пуассоновский точечный процесс (иначе называемый процессом Кокса), управляемый некоторым процессом $\Lambda(t)$. А именно, если $N(t) = \max\{j :\ Y_j \le t\}$, $t \ge 0$ — общее число модификаций до момента t, то в рассматриваемом случае

$$N(t) = N_1(\Lambda(t))\,,$$

где N_1 — стандартный пуассоновский процесс (однородный пуассоновский процесс с единичной интенсивностью), независимый от процесса $\Lambda(t)$, траектории которого выходят из нуля, не убывают, непрерывны справа и почти наверное конечны. Всюду в дальнейшем будет предполагаться, что

$\mathsf{E}\Lambda(t) \equiv t$. Это предположение можно интерпретировать и как то, что управляющий процесс в среднем пропорционален времени, и (что существенно важно для построения предельных аппроксимаций) как то, что задача параметризована математическим ожиданием управляющего процесса.

Заметим, что в используемых обозначениях

$$P(Y_{j+1}-0) = P(Y_j)\,, \quad j \geq 1\,,$$

и

$$P(t) = P(Y_{N(t)})\,, \quad t > 0\,. \tag{1}$$

2 Неоднородные экспоненциальные модели с непрерывным временем

Пусть $\{(\theta_j, \eta_j)\}_{j\geq 1}$ — последовательность независимых одинаково распределенных двумерных случайных векторов таких, что

$$0 \leq \theta_1 \leq 1\,, \;\; 0 \leq \eta_1 \leq 1 \;\text{ почти наверное}\,.$$

Отметим, что независимость θ_j и η_j внутри каждой пары, равно как и совпадение распределений θ_j и η_j внутри каждой пары, не предполагается. Однако будем считать, что последовательность пар $\{(\theta_j, \eta_j)\}_{j\geq 1}$ стохастически независима от точечного случайного процесса $Y_1, Y_2, \ldots$

Задав начальную надежность p_0, рассмотрим модель, определяемую рекуррентным соотношением

$$P(Y_{j+1}) = \theta_{j+1}P(Y_j) + \eta_{j+1}\big(1-P(Y_j)\big)\,, \;\; j \geq 0\,. \tag{2}$$

Эту модель назовем неоднородной *экспоненциальной* моделью с непрерывным временем. В такой модели случайные величины θ_j описывают возможное уменьшение надежности из-за некачественных модификаций, в ходе которых вместо исправления существующих дефектов в систему могут быть внесены новые, в то время как величины η_j описывают повышение надежности за счет исправления дефектов. Частные случаи модели (2) с двухточечными распределениями случайных величин θ_j и η_j и дискретным временем рассматривались в [5, 6]. В свою очередь, эти частные случаи представляют собой переформулировку в терминах теории надежности одной модели обучаемости, рассмотренной в [7]. Таким образом, можно пополнить список различных приложений моделей, рассматриваемых в данной статье, еще и теорией обучаемости.

Обозначим $\mathsf{E}\theta_1 = 1-a$, $\mathsf{E}\eta_1 = b$.

Теорема 1. *Для любого* $t > 0$

$$\mathsf{E}P(t) = \frac{b}{a+b} + \Big(p_0 - \frac{b}{a+b}\Big)\mathsf{E}e^{-(a+b)\Lambda(t)}.$$

Доказательство. Сначала заметим, что

$$\mathsf{E}P(Y_j) = \frac{b}{a+b} + \\ + \left(p_0 - \frac{b}{a+b}\right)(1-a-b)^j\,, \;\; j \geq 1\,. \tag{3}$$

Действительно, взяв математические ожидания от обеих частей соотношения (2) с учетом независимости векторов (θ_j, η_j), получим

$$\mathsf{E}P(Y_{j+1}) = \mathsf{E}P(Y_j)(1-a-b) + b\,. \tag{4}$$

Разрешая рекурсию (4), получаем соотношение (3).

Теперь, воспользовавшись соотношениями (1) и (3), по формуле полной вероятности получим

$$\mathsf{E}P(t) = \sum_{j=0}^{\infty} \mathsf{E}\big[P(t)|N(t)=j\big]\mathsf{P}\big(N(t)=j\big) = \\ = \sum_{j=0}^{\infty} \mathsf{E}P(Y_j)\mathsf{P}\big(N(t)=j\big) = \\ = \sum_{j=0}^{\infty} \mathsf{E}P(Y_j) \int_0^{\infty} e^{-\lambda}\frac{\lambda^j}{j!}\,d\mathsf{P}\big(\Lambda(t)<\lambda\big) = \\ = \int_0^{\infty} e^{-\lambda} \sum_{j=0}^{\infty} \frac{\lambda^j}{j!}\Big[\frac{b}{a+b} + \\ + \Big(p_0 - \frac{b}{a+b}\Big)(1-a-b)^j\Big]\,d\mathsf{P}\big(\Lambda(t)<\lambda\big) = \\ = \frac{b}{a+b} + \Big(p_0 - \frac{b}{a+b}\Big) \times \\ \times \int_0^{\infty} \left(e^{-\lambda}\sum_{j=0}^{\infty}\frac{[\lambda(1-a-b)]^j}{j!}\right) d\mathsf{P}\big(\Lambda(t)<\lambda\big) = \\ = \frac{b}{a+b} + \Big(p_0 - \frac{b}{a+b}\Big)\int_0^{\infty} e^{-(a+b)\lambda}\,d\mathsf{P}\big(\Lambda(t)<\lambda\big) = \\ = \frac{b}{a+b} + \Big(p_0 - \frac{b}{a+b}\Big)\mathsf{E}e^{-(a+b)\Lambda(t)}\,.$$

Теорема доказана.

Из теоремы 1 вытекает, что если $\mathsf{E}e^{-(a+b)\Lambda(t)} \to 0$ при $t \to \infty$ (что может иметь место, если, к примеру, $\Lambda(t) \to \infty$ по вероятности при $t \to \infty$), то в зависимости от знака величины $c = (a+b)p_0 - b$ ожидаемая надежность системы либо возрастает (если $c < 0$), либо убывает (если $c > 0$). Однако в любом случае справедливо

Следствие 1. *Пусть* $\Lambda(t) \to \infty$ *по вероятности при* $t \to \infty$. *Тогда*

$$\lim_{t\to\infty} \mathsf{E}P(t) = \frac{b}{a+b}\,.$$

В силу неотрицательности и ограниченности случайных величин θ_j из следствия 1, в свою очередь, вытекает

Следствие 2. *Пусть* $\Lambda(t) \to \infty$ *по вероятности при* $t \to \infty$. *Соотношение*

$$\lim_{t\to\infty} \mathsf{E}P(t) = 1$$

имеет место в том и только в том случае, когда $\mathsf{P}(\theta_1 = 1)$.

Другими словами, в рамках экспоненциальной модели абсолютная надежность может быть достигнута только за счет идеально правильных модификаций, в ходе которых полностью исключена возможность внесения каких-либо новых дефектов.

Следствие 3. *Пусть* $\Lambda(t) \equiv t$, *т. е.* $N(t)$ — *стандартный пуассоновский процесс. Тогда для любого* $t > 0$

$$\mathsf{E}P(t) = \frac{b}{a+b} + \Big(p_0 - \frac{b}{a+b}\Big)e^{-(a+b)t}\,.$$

Рассмотрим ситуацию, когда $\theta_j = 1$ почти наверное, более подробно. В этом случае соотношение (2) можно переписать в виде

$$1 - P(Y_{j+1}) = (1-\eta_{j+1})\big(1 - P(Y_j)\big)\,,\ \ j \geq 1\,. \quad (5)$$

Обозначим $\log(1-\eta_j) = \zeta_j$. Тогда из (5) получаем

$$\log\big(1 - P(t)\big) - \log(1-p_0) = \sum_{k=1}^{N(t)} \zeta_k\,. \quad (6)$$

В правой части представления (6) стоит обобщенный дважды стохастический пуассоновский процесс (обобщенный процесс Кокса). Это обстоятельство позволяет воспользоваться предельными теоремами для обобщенных процессов Кокса (см., например, [8] или [9]) для уточнения асимптотического поведения надежности системы в рамках неоднородной экспоненциальной модели с непрерывным временем.

Предположим, что $0 < \mathsf{D}\zeta_j = \sigma^2 < \infty$, и обозначим $\mathsf{E}\zeta_j = \alpha$. Заметим, что $\alpha \leq 0$, так как $0 \leq \eta_j \leq 1$. Сходимость по распределению (слабую сходимость) будем обозначать символом $\Longrightarrow$. Пусть $\Phi(x)$ — стандартная нормальная функция распределения. В дополнение к условию $\mathsf{E}\Lambda(t) \equiv t$, введенному выше, предположим, что $\mathsf{D}\Lambda(t) \equiv s^2 t$ для некоторого $s \in [0, \infty)$. Справедлив следующий результат.

Теорема 2. *Предположим, что* $\alpha \neq 0$, $\mathsf{E}\Lambda(t) \equiv t$, $\mathsf{D}\Lambda(t) \equiv s^2 t$ *для некоторого* $s \in [0, \infty)$ *и* $\Lambda(t) \xrightarrow{P} \infty$ *при* $t \to \infty$. *Тогда одномерные распределения неслучайно центрированного и нормированного случайного процесса* $P(t)$ *слабо сходятся к распределению некоторой случайной величины* Z *при* $t \to \infty$, *т. е.*

$$\frac{\log\big(1-P(t)\big) - \log(1-p_0) - \alpha t}{\sqrt{[\alpha^2(1+s^2)+\sigma^2]t}} \Longrightarrow Z \ \ (t \to \infty)\,,$$

тогда и только тогда, когда существует случайная величина V *такая, что*

$$\frac{\Lambda(t) - t}{s\sqrt{t}} \Longrightarrow V \ \ (t \to \infty)\,.$$

При этом

$$\mathsf{P}(Z < x) = \\ = \mathsf{E}\Phi\left(x\sqrt{1 + \frac{\alpha^2 s^2}{\alpha^2+\sigma^2}} - \frac{\alpha s V}{\sqrt{\sigma^2+\alpha^2}}\right)\,,\ \ x \in \mathbb{R}\,.$$

Несложно видеть, что предельная случайная величина Z допускает представление

$$Z \stackrel{d}{=} \left[1 + \frac{\alpha^2 s^2}{\alpha^2+\sigma^2}\right]^{-1/2} X + \frac{\alpha s}{\sqrt{a^2(1+s^2)+\sigma^2}}\,V\,,$$

где X — случайная величина со стандартным нормальным распределением, независимая от случайной величины V.

Теорема 2 является просто иной формой записи теоремы 9.2.2 из [8].

Для построения «гарантированно доверительных» границ для надежности системы в рамках неоднородной экспоненциальной модели можно воспользоваться оценками скорости сходимости в теореме 9.2.2, полученными в статье [10].

Дополнительно предположим, во-первых, что $\mathsf{E}|\zeta_1|^3 < \infty$, и обозначим

$$\beta^3 = \mathsf{E}|\zeta_1|^3\,,\ \ L_3 = \frac{\beta^3}{(\alpha^2+\sigma^2)^{3/2}}\,.$$

Во-вторых, предположим, что случайная величина V, фигурирующая в теореме 2, имеет стандартное нормальное распределение, причем выполнено соотношение

$$\sup_x \left|\mathsf{P}\left(\frac{\Lambda(t)-t}{s\sqrt{t}} < x\right) - \Phi(x)\right| \leq \frac{\kappa}{\sqrt{t}} \quad (7)$$

при некотором $\kappa \in (0, \infty)$. Соотношение (7) выполнено, например, если $\Lambda(t)$ имеет гамма-распределение с параметром масштаба, равным s, и параметром формы, равным t (заметим, что при этом случайная величина $N(t)$ имеет отрицательное биномиальное распределение).

Для такого случая распределение случайной величины Z является стандартным нормальным и оценка, полученная в работе [10], имеет вид

$$\sup_x \left| \mathsf{P}\left(\frac{\log\big(1-P(t)\big) - \log(1-p_0) - \alpha t}{\sqrt{[\alpha^2(1+s^2)+\sigma^2]t}} < x \right) - \Phi(x) \right| \le \frac{1}{\sqrt{t}} \left[\kappa + \inf_{\epsilon \in (0,1)} \left\{ \frac{C_0 L_3}{\sqrt{1-\epsilon}} + \frac{s\sqrt{2}}{\sqrt{\pi}\epsilon} + sM(\epsilon) \right\} \right], \quad (8)$$

где C_0 — абсолютная постоянная в неравенстве Берри–Эссеена, $C_0 \le 0{,}7005$ (см. работу [11]),

$$M(\epsilon) = \max\left\{ \frac{1}{\epsilon}, \frac{\sqrt{1+\epsilon}}{\big(1+\sqrt{1-\epsilon}\big)\sqrt{2\pi e(1-\epsilon)}} \right\}.$$

Обозначим

$$K = K(L_3, s^2, \kappa) = \kappa + \inf_{\epsilon \in (0,\,1)} \left\{ \frac{C_0 L_3}{\sqrt{1-\epsilon}} + \frac{s\sqrt{2}}{\sqrt{\pi}\epsilon} + sM(\epsilon) \right\}. \quad (9)$$

Тогда из (8) для $z \in (0,\,1)$ вытекает справедливость неравенств

$$\Phi\left(\frac{\log(1-z) - \log(1-p_0) - \alpha t}{\sqrt{t[\alpha^2(1+s^2)+\sigma^2]}} \right) - \frac{K}{\sqrt{t}} \le \le \mathsf{P}\big(P(t) > z\big) \le \le \Phi\left(\frac{\log(1-z) - \log(1-p_0) - \alpha t}{\sqrt{t[\alpha^2(1+s^2)+\sigma^2]}} \right) + \frac{K}{\sqrt{t}}.$$

Зададим коэффициент доверия $\gamma \in (1/2,\,1)$ и решим относительно z уравнение

$$\Phi\left(\frac{\log(1-z) - \log(1-p_0) - \alpha t}{\sqrt{t[\alpha^2(1+s^2)+\sigma^2]}} \right) - \frac{K}{\sqrt{t}} = \gamma.$$

Получим нижнюю гарантированную доверительную границу $\underline{z}_\gamma(t)$ для $P(t)$ с коэффициентом доверия γ:

$$\underline{z}_\gamma(t) = 1 - \exp\left\{ \alpha t + \sqrt{t[\alpha^2(1+s^2)+\sigma^2]}\, u\left(\gamma + \frac{K}{\sqrt{t}}\right) + \log(1-p_0) \right\},$$

где для $v \in (0,\,1)$ символом $u(v)$ обозначается v-квантиль стандартного нормального распределения. При этом для каждого $t > 0$

$$\mathsf{P}\big(P(t) > \underline{z}_\gamma(t)\big) \ge \gamma.$$

Чтобы построить двусторонние границы для $P(t)$, заметим, что из (8) вытекает неравенство

$$\sup_x \left| \mathsf{P}\left(\left| \frac{\log\big(1-P(t)\big) - \log(1-p_0) - \alpha t}{\sqrt{[\alpha^2(1+s^2)+\sigma^2]t}} \right| < x \right) - 2\Phi(x) + 1 \right| \le \frac{2K}{\sqrt{t}}.$$

Отсюда получается гарантированная доверительная полоса $\big\{\big(z_\gamma^{(1)}(t),\, z_\gamma^{(2)}(t)\big) : t > 0\big\}$ для $P(t)$ с коэффициентом доверия γ, где

$$z_\gamma^{(1)}(t) = 1 - \exp\left\{ \alpha t + u\frac{1}{2}\left(\gamma + 1 + 2\frac{K}{\sqrt{t}} \right) \sqrt{t[\alpha^2(1+s^2)+\sigma^2]} + \log(1-p_0) \right\};$$

$$z_\gamma^{(2)}(t) = 1 - \exp\left\{ \alpha t - u\frac{1}{2}\left(\gamma + 1 + 2\frac{K}{\sqrt{t}} \right) \sqrt{t[\alpha^2(1+s^2)+\sigma^2]} + \log(1-p_0) \right\}.$$

При этом для каждого $t > 0$

$$\mathsf{P}\big(z_\gamma^{(1)}(t) \le P(t) \le z_\gamma^{(2)}(t)\big) \ge \gamma.$$

3 Неоднородные логистические модели с непрерывным временем

Обозначим

$$Q(Y_j) = \frac{P(Y_j)}{1 - P(Y_j)}, \quad j \ge 1.$$

Пусть $\theta_1, \theta_2, \ldots$ — независимые одинаково распределенные случайные величины. Будем считать, что последовательность $\{\theta_j\}_{j \ge 1}$ стохастически независима от точечного случайного процесса $Y_1, Y_2, \ldots$

Предположим, что

$$Q(Y_{j+1}) = \theta_{j+1} Q(Y_j), \quad j \ge 0. \quad (10)$$

Эту модель изменения надежности, называемую *логистической*, можно интерпретировать следующим образом. Если $p_j = P(Y_j)$ — вероятность успеха

в последовательности испытаний Бернулли, то величина $q_j = p_j/(1-p_j)$ характеризует ожидаемый номер испытания, заканчивающегося первой неудачей в этой последовательности испытаний Бернулли. Таким образом, величина $Q(Y_j) = q_j$ характеризует ожидаемое время жизни (безотказной работы) системы после j-й модификации, как если бы после момента Y_j в нее не вносились бы никакие изменения. Следовательно, соотношение (10) можно интерпретировать как формализацию того, что каждая модификация системы изменяет ожидаемое время безотказной работы после модификации в случайное число раз. Положим

$$Q(t) = Q(Y_{N(t)}), \quad t > 0.$$

Обозначим $\mathsf{E}\theta_1 = a$. Пусть p_0 — надежность системы в момент $t = 0$. Обозначим

$$q_0 = \frac{p_0}{1-p_0}.$$

Теорема 3. *В рамках неоднородной логистической модели справедливо соотношение*

$$\mathsf{E}Q(t) = q_0\mathsf{E}e^{(a-1)\Lambda(t)}, \quad t > 0.$$

Доказательство. Из соотношения (10) почти очевидно, что

$$\mathsf{E}Q(Y_j) = q_0 a^j, \quad j \geq 1.$$

Поэтому по формуле полной вероятности

$$\begin{aligned}\mathsf{E}Q(t) &= \sum_{j=0}^{\infty} \mathsf{E}Q(Y_j)\mathsf{P}\big(N(t)=j\big) = \\ &= \sum_{j=0}^{\infty} \int_0^{\infty} e^{-\lambda}\frac{\lambda^j}{j!}a^j q_0\, d\mathsf{P}\big(\Lambda(t)<\lambda\big) = \\ &= q_0 \int_0^{\infty} e^{-\lambda}\left(\sum_{j=0}^{\infty}\frac{(a\lambda)^j}{j!}\right) d\mathsf{P}\big(\Lambda(t)<\lambda\big) = \\ &= q_0 \int_0^{\infty} e^{\lambda(a-1)}\, d\mathsf{P}\big(\Lambda(t)<\lambda\big) = q_0\mathsf{E}e^{(a-1)\Lambda(t)}.\end{aligned}$$

Теорема доказана.

Так как функция

$$g(x) = \frac{x}{1-x}$$

выпукла при $0 \leq x < 1$, то с помощью неравенства Иенсена из теоремы 3 легко получить

Следствие 4. *В рамках неоднородной логистической модели справедливо соотношение*

$$\mathsf{E}P(t) \leq \frac{q_0\mathsf{E}e^{(a-1)\Lambda(t)}}{1+q_0\mathsf{E}e^{(a-1)\Lambda(t)}}, \quad t > 0.$$

Следствие 5. *Если $N(t)$ — стандартный пуассоновский процесс, то в рамках неоднородной логистической модели при любом $t > 0$ справедливы соотношения*

$$\mathsf{E}Q(t) = q_0\mathsf{E}e^{(a-1)t}, \quad \mathsf{E}P(t) \leq \frac{q_0 e^{(a-1)t}}{1+q_0 e^{(a-1)t}}.$$

Обозначим $\log\theta_j = \chi_j$. Тогда из (10) получаем соотношение

$$\log Q(Y_j) - \log q_0 = \sum_{k=1}^{j}\chi_k,$$

откуда вытекает, что

$$\log Q(t) - \log q_0 = \sum_{k=1}^{N(t)}\chi_k.$$

Таким образом, по аналогии с теоремой 2 легко получить следующее утверждение. Предположим, что $0 < \mathsf{D}\chi_j \equiv \sigma^2 < \infty$, и обозначим $\mathsf{E}\chi_j = \alpha$.

Теорема 4. *Предположим, что $\alpha \neq 0$, $\mathsf{E}\Lambda(t) \equiv t$, $\mathsf{D}\Lambda(t) \equiv s^2 t$ для некоторого $s \in [0,\infty)$ и $\Lambda(t) \xrightarrow{P} \infty$ при $t \to \infty$. Тогда одномерные распределения неслучайно центрированного и нормированного случайного процесса $Q(t)$ слабо сходятся к распределению некоторой случайной величины Z при $t \to \infty$, т. е.*

$$\frac{\log Q(t) - \log q_0 - \alpha t}{\sqrt{[\alpha^2(1+s^2)+\sigma^2]t}} \Longrightarrow Z \quad (t \to \infty),$$

тогда и только тогда, когда существует случайная величина V такая, что

$$\frac{\Lambda(t)-t}{s\sqrt{t}} \Longrightarrow V \quad (t \to \infty).$$

При этом

$$\begin{aligned}&\mathsf{P}(Z<x) = \\ &= \mathsf{E}\Phi\left(x\sqrt{1+\frac{\alpha^2 s^2}{\alpha^2+\sigma^2}} - \frac{\alpha s V}{\sqrt{\sigma^2+\alpha^2}}\right), \quad x \in \mathbb{R}.\end{aligned}$$

Снова предположим, что имеет место соотношение (7). Для унификации обозначений предположим, что $\mathsf{E}|\chi_1|^3 < \infty$, и обозначим

$$\beta^3 = \mathsf{E}|\chi_1|^3, \quad L_3 = \frac{\beta^3}{(\alpha^2+\sigma^2)^{3/2}}.$$

Из результатов работы [10] вытекает, что в сделанных предположениях справедлива оценка

$$\sup_x \left| \mathsf{P}\left(\frac{\log Q(t) - \log q_0 - \alpha t}{\sqrt{[\alpha^2(1+s^2)+\sigma^2]t}} < x \right) - \Phi(x) \right| \le \le \frac{K}{\sqrt{t}}, \quad (11)$$

где величина K определена в предыдущем разделе (см. (9)). Эта оценка позволяет получить гарантированную нижнюю доверительную границу для надежности системы $P(t)$ в рамках модели (10), которая для заданного коэффициента доверия $\gamma \in [1/2,\, 1)$ имеет вид

$$\underline{z}_\gamma(t) = \frac{\exp\{\underline{x}_\gamma(t)\}}{1+\exp\{\underline{x}_\gamma(t)\}},$$

где

$$\underline{x}_\gamma(t) = \alpha t + \\ + u\left(1-\gamma-\frac{K}{\sqrt{t}}\right)\sqrt{t[\alpha^2(1+s^2)+\sigma^2]} + \log q_0 .$$

При этом для каждого $t > 0$

$$\mathsf{P}\big(P(t) > \underline{z}_\gamma(t)\big) \ge \gamma .$$

Чтобы получить двусторонние гарантированные границы для надежности системы $P(t)$ в рамках модели (10), заметим, что из оценки (11) вытекает неравенство

$$\sup_x \left| \mathsf{P}\left(\left| \frac{\log Q(t) - \log q_0 - \alpha t}{\sqrt{[\alpha^2(1+s^2)+\sigma^2]t}} \right| < x \right) - \\ - 2\Phi(x) + 1 \right| \le \frac{2K}{\sqrt{t}} .$$

Из этого неравенства следует, что гарантированная доверительная полоса

$$\big\{ \big(z_\gamma^{(1)}(t),\, z_\gamma^{(2)}(t)\big) :\, t > 0 \big\}$$

для $P(t)$ с коэффициентом доверия γ в рамках логистической модели (10) имеет вид

$$z_\gamma^{(1)}(t) = \frac{\exp\left\{\alpha t - u\left((1/2)(\gamma+1+2K/\sqrt{t})\right)\sqrt{t[\alpha^2(1+s^2)+\sigma^2]} + \log q_0\right\}}{1+\exp\left\{\alpha t - u\left((1/2)(\gamma+1+2K/\sqrt{t})\right)\sqrt{t[\alpha^2(1+s^2)+\sigma^2]} + \log q_0\right\}},$$

$$z_\gamma^{(2)}(t) = \frac{\exp\left\{\alpha t + u\left((1/2)(\gamma+1+2K/\sqrt{t})\right)\sqrt{t[\alpha^2(1+s^2)+\sigma^2]} + \log q_0\right\}}{1+\exp\left\{\alpha t + u\left((1/2)(\gamma+1+2K/\sqrt{t})\right)\sqrt{t[\alpha^2(1+s^2)+\sigma^2]} + \log q_0\right\}}.$$

При этом для каждого $t > 0$

$$\mathsf{P}\big(z_\gamma^{(1)}(t) \le P(t) \le z_\gamma^{(2)}(t)\big) \ge \gamma .$$

4 Неоднородные гиперболические модели с непрерывным временем

В исходных обозначениях и предположениях предыдущего раздела рассмотрим модель

$$Q(Y_{j+1}) = Q(Y_j) + \theta_{j+1}, \;\; j \ge 0 . \quad (12)$$

Эту модель изменения надежности, называемую *гиперболической*, можно интерпретировать как формализацию того, что каждая модификация системы изменяет ожидаемое время ее безотказной работы после модификации на случайное время. Как и ранее, полагаем

$$Q(t) = Q(Y_{N(t)}), \quad t > 0 .$$

Обозначим $\mathsf{E}\theta_1 = \alpha$. Пусть, как и ранее, p_0 — надежность системы в момент $t = 0$, $q_0 = p_0/(1-p_0)$.

Из соотношения (12) очевидным образом вытекает, что в рамках неоднородной гиперболической модели

$$\mathsf{E}Q(t) = q_0 + \alpha t, \quad t > 0 .$$

Отсюда с помощью неравенства Иенсена получаем неравенство

$$\mathsf{E}P(t) \le \frac{q_0 + \alpha t}{1 + q_0 + \alpha t},$$

справедливое при любом $t > 0$.

Предположим теперь, что $0 < \mathsf{D}\theta_1 \equiv \sigma^2 < \infty$. Из рекуррентного соотношения (12) следует, что

$$Q(t) = q_0 + \sum_{j=0}^{N(t)} \theta_j, \quad t > 0 .$$

Таким образом, для уточнения асимптотики надежности $P(t)$ системы в рамках гиперболической модели опять можно воспользоваться предельными теоремами для обобщенных процессов Кокса.

Теорема 5. *Предположим, что* $\alpha \neq 0$, $\mathsf{E}\Lambda(t) \equiv t$, $\mathsf{D}\Lambda(t) \equiv s^2 t$ *для некоторого* $s \in [0, \infty)$ *и* $\Lambda(t) \xrightarrow{P} \infty$ *при* $t \to \infty$. *Тогда одномерные распределения неслучайно центрированного и нормированного случайного процесса* $Q(t)$ *слабо сходятся к распределению некоторой случайной величины* Z *при* $t \to \infty$, *т. е.*

$$\frac{Q(t) - q_0 - \alpha t}{\sqrt{[\alpha^2(1+s^2)+\sigma^2]t}} \Longrightarrow Z \quad (t \to \infty)\,,$$

тогда и только тогда, когда существует случайная величина V *такая, что*

$$\frac{\Lambda(t) - t}{s\sqrt{t}} \Longrightarrow V \quad (t \to \infty)\,.$$

При этом

$$\mathsf{P}(Z < x) = \\ = \mathsf{E}\Phi\left(x\sqrt{1 + \frac{\alpha^2 s^2}{\alpha^2+\sigma^2}} - \frac{\alpha s V}{\sqrt{\sigma^2+\alpha^2}}\right), \quad x \in \mathbb{R}\,.$$

Снова предположим, что имеет место соотношение (7). Для унификации обозначений предположим, что $\mathsf{E}|\theta_1|^3 < \infty$, и обозначим

$$\beta^3 = \mathsf{E}|\theta_1|^3\,, \quad L_3 = \frac{\beta^3}{(\alpha^2+\sigma^2)^{3/2}}\,.$$

Из результатов работы [10] вытекает, что в сделанных предположениях справедлива оценка

$$\sup_x \left|\mathsf{P}\left(\frac{Q(t) - q_0 - \alpha t}{\sqrt{[\alpha^2(1+s^2)+\sigma^2]t}} < x\right) - \Phi(x)\right| \leq \\ \leq \frac{K}{\sqrt{t}}\,, \quad (13)$$

где величина K определена в соотношении (9). Эта оценка позволяет получить гарантированную нижнюю доверительную границу для надежности системы $P(t)$ в рамках модели (12), которая для заданного коэффициента доверия $\gamma \in [1/2, 1)$ имеет вид

$$\underline{z}_\gamma(t) = \frac{\underline{x}_\gamma(t)}{1 + \underline{x}_\gamma(t)}\,,$$

где

$$\underline{x}_\gamma(t) = \\ = \alpha t + u\left(1 - \gamma - \frac{K}{\sqrt{t}}\right)\sqrt{t[\alpha^2(1+s^2)+\sigma^2]} + q_0\,.$$

При этом для каждого $t > 0$

$$\mathsf{P}\big(P(t) > \underline{z}_\gamma(t)\big) \geq \gamma\,.$$

Чтобы получить двусторонние гарантированные границы для надежности системы $P(t)$ в рамках модели (12), заметим, что из оценки (13) вытекает неравенство

$$\sup_x \left|\mathsf{P}\left(\left|\frac{Q(t) - q_0 - \alpha t}{\sqrt{[\alpha^2(1+s^2)+\sigma^2]t}}\right| < x\right) - \\ - 2\Phi(x) + 1\right| \leq \frac{2K}{\sqrt{t}}\,.$$

Из этого неравенства следует, что гарантированная доверительная полоса

$$\big\{\big(z_\gamma^{(1)}(t),\, z_\gamma^{(2)}(t)\big) :\, t > 0\big\}$$

для $P(t)$ с коэффициентом доверия γ в рамках гиперболической модели (12) имеет вид

$$z_\gamma^{(1)}(t) = \frac{\alpha t - u\left((1/2)(\gamma + 1 + 2K/\sqrt{t})\right)\sqrt{t[\alpha^2(1+s^2)+\sigma^2]} + q_0}{1 + \alpha t - u\left((1/2)(\gamma + 1 + 2K/\sqrt{t})\right)\sqrt{t[\alpha^2(1+s^2)+\sigma^2]} + q_0}\,,$$

$$z_\gamma^{(2)}(t) = \frac{\alpha t + u\left((1/2)(\gamma + 1 + 2K/\sqrt{t})\right)\sqrt{t[\alpha^2(1+s^2)+\sigma^2]} + q_0}{1 + \alpha t + u\left((1/2)(\gamma + 1 + 2K/\sqrt{t})\right)\sqrt{t[\alpha^2(1+s^2)+\sigma^2]} + q_0}\,.$$

При этом для каждого $t > 0$

$$\mathsf{P}\big(z_\gamma^{(1)}(t) \leq P(t) \leq z_\gamma^{(2)}(t)\big) \geq \gamma\,.$$

Литература

1. *Gnedenko B. V., V. Yu. Korolev*. Random summation: Limit theorems and applications. — Boca Raton: CRC Press, 1996.
2. *Королёв В. Ю.* Прикладные задачи теории вероятностей: модели роста надежности модифицируемых систем. — М.: Диалог-МГУ, 1997.
3. *Королёв В. Ю., Соколов И. А.* Основы математической теории надежности модифицируемых систем. — М.: Изд-во ИПИРАН, 2006.
4. *Бенинг В. Е., Королёв В. Ю., Соколов И. А., Шоргин С. Я.* Рандомизированные модели и методы теории надежности информационных и технических систем. — М.: Торус Пресс, 2007.
5. *Волков Л. И., Шишкевич А. М.* Надежность летательных аппаратов. — М.: Высшая школа, 1975.
6. *Волков Л. И.* Управление эксплуатацией летательных комплексов. — М.: Высшая школа, 1981.
7. *Буш Р., Мостеллер Ф.* Стохастические модели обучаемости. — М.: ГИФМЛ, 1962.
8. *Bening V. E., Korolev V. Yu.* Generalized Poisson models and their applications in insurance and finance. — Utrecht: VSP, 2002.
9. *Бенинг В. Е., Королёв В. Ю., Шоргин С. Я.* Математические основы теории риска. — М.: Физматлит, 2007.
10. *Артюхов С. В., Королёв В. Ю.* Оценки скорости сходимости распределений обобщенных дважды стохастических пуассоновских процессов с ненулевым средним // Обозрение промышленной и прикладной математики, 2008 (в печати).
11. *Шевцова И. Г.* Об абсолютной постоянной в неравенстве Берри–Эссеена // Сб. статей молодых ученых факультета ВМиК МГУ. — М.: Изд-во факультета ВМиК МГУ, 2008. Вып. 5. С. 101–110.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2008. Т. 2. Вып. 4. С. 66–73

ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ ИНТЕГРАЦИИ ИДЕНТИФИКАЦИИ ПО ИЗОБРАЖЕНИЮ ЛИЦА ДЛЯ УСКОРЕНИЯ АВТОМАТИЧЕСКОЙ ДАКТИЛОСКОПИЧЕСКОЙ ИДЕНТИФИКАЦИИ

О. С. Ушмаев[1]

Аннотация: Рассмотрена проблема синтеза мультибиометрических систем распознавания по отпечаткам пальцев и изображению лица. Основное внимание уделено производительности. Предложены методы организации вычислений, оптимизирующие скорость сравнения биометрических образцов. Проведенные эксперименты показывают эффективность разработанных методов.

Ключевые слова: биометрические технологии; мультибиометрическая идентификация; идентификация по отпечаткам пальцев; идентификация по изображению лица; оптимизация производительности

1 Введение

К настоящему времени накоплен значительный опыт создания крупномасштабных биометрических систем. Большинство из них являются автоматизированными дактилоскопическими системами (АДИС). Выбор отпечатков пальцев обусловлен множеством факторов. Основным из них является традиционное использование отпечатков пальцев в криминальном учете и высокий потенциал дактилоскопии с точки зрения ошибок распознавания. Использование для распознавания всех 10 отпечатков пальцев достаточно для практически безошибочного распознавания людей в масштабах населения страны. При использовании меньшего числа отпечатков соотношение ошибок 1-го (FRR — False Rejection Rate) и 2-го (FAR — False Acceptance Rate) родов является удовлетворительных во многих приложениях (рис. 1) [1]. Однако использование дактилоскопии имеет недостатки. Накопленный опыт реализации АДИС для криминальной и гражданской идентификации позволяет выделить следующие направления развития [2–4]:

- идентификация людей, не обладающих пригодными для распознавания отпечатками пальцев (инвалиды, плохое состояние кожи);
- увеличение производительности.

Первая задача возникает из непосредственных требований к биометрическим системам, а именно она должна автоматически идентифицировать личность по предъявляемым биометрическим образцам. Соответственно люди с плохим качеством отпечатков пальцев не могут надежно идентифицироваться средствами АДИС.

Актуальность увеличения производительности в первую очередь связана с тем, что в настоящее время при создании крупномасштабных биометрических систем более 75% средств затрачиваются на аппаратные средства вычислительных узлов. При полномасштабном внедрении систем гражданской идентификации таких, как паспортно-визовые документы нового поколения, биометрические массивы вырастут многократно, поэтому эффективное решение задачи увеличения производительности может значительно повысить эффективность внедрения биометрии.

Обе указанные задачи могут быть решены путем добавления дополнительного биометрического идентификатора [5]. Наиболее доступной и удобной дополнительной биометрикой является изображение лица. В частности, в крупномасштабных системах дактилоскопической идентификации (криминальные учеты, паспортно-визовые системы) помимо отпечатков пальцев доступна фотография как традиционный способ идентификации личности.

С технической точки зрения, лицевая биометрия потенциально может быть использована для решения задачи увеличения производительности АДИС, поскольку скорость идентификации по изображению лица многократно выше скорости

[1]Институт проблем информатики Российской академии наук, oushmaev@ipiran.ru

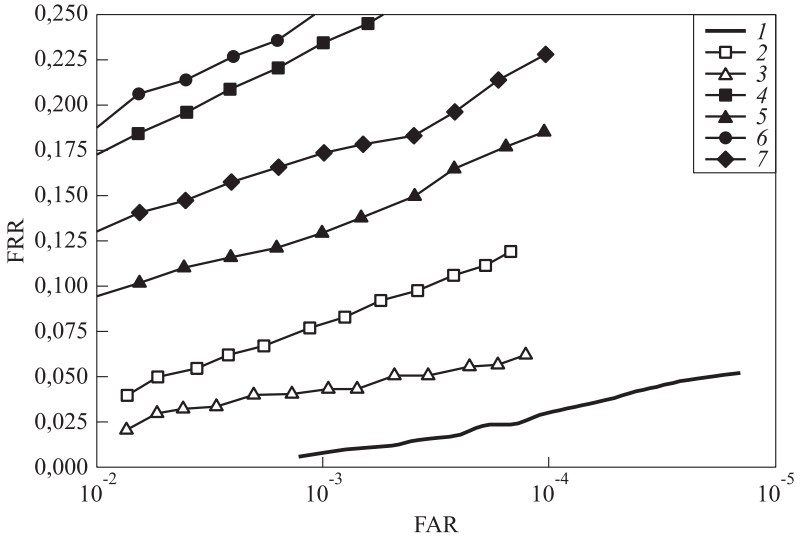


Рис. 1 Ошибки 1-го и 2-го рода распознавания по нескольким отпечаткам пальцев (NIST SD14) [2]: *1* — 4 отпечатка, *2* — 2 отпечатка (указательные), *3* — 2 отпечатка (большие), *4* — правый указательный, *5* — правый большой, *6* — левый указательный, *7* — левый большой

сравнения по отпечаткам пальцев. Данное обстоятельство позволяет использовать результаты сравнения по изображению лица для грубого «отсева» части субъектов, что сократит нагрузку на вычислительные узлы АДИС.

Далее в статье рассмотрена задача увеличения производительности АДИС без потери качества распознавания за счет добавления лицевой биометрии. В разделе 2 дана методология оценки производительности биометрической идентификации. Алгоритмы интеграции идентификации по изображению лица в АДИС изложены в разделе 3. В разделе 4 представлены теоретические оценки производительности интегрированной мультибиометрической системы распознавания по отпечаткам пальцев и изображению лица. В разделе 5 приведены результаты экспериментов по моделированию изменения производительности. В разделе 6 представлены основные выводы и заключение.

2 Методология оценки производительности

Основным показателем производительности биометрической системы является проектное время ожидание отклика. В случае крупномасштабных систем, таких как паспортно-визовые системы или системы криминального учета, время отклика обычно устанавливается в 24 ч для суточного цикла (в редких случаях — 168 ч для недельного цикла) функционирования системы. Соответственно, биометрическая система должна успевать обрабатывать заявки на идентификацию, поступающие в течение суток. Это условие слабее требования реального времени. Его использование связано с тем, что поток заявок имеет прогнозируемую неравномерную структуру. Примерный вид графика интенсивности запроса приведен на рис. 2. Максимальный участок соответствует времени, когда функционирует большинство пунктов сбора биометрической информации во всех часовых поясах.

В таком случае проектное время обработки заявок, поступающих в течение суток, рассчитывается по следующей формуле:

$$T_{\text{сут}} = r_{\text{сут}} t\,.$$

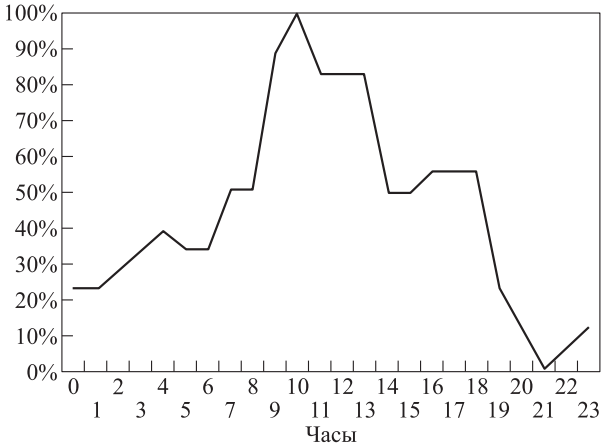


Рис. 2 Примерный график интенсивности запросов

Здесь $r_{\text{сут}}$ — максимальный проектный поток заявок в течение суток; t — среднее время идентификации по биометрической базе, в случае АДИС (или другой однофакторной биометрии) обычно линейно пропорционально количеству записей в базе данных (БД), поскольку в ходе идентификации предъявляемые образцы последовательно сравниваются с каждым хранимым, т. е. в большинстве приложений

$$t = \frac{Nt_{\text{ср}}}{W}\,, \qquad (1)$$

где N — число записей в БД; $t_{\text{ср}}$ — среднее время сравнения пары биометрических образцов на единицу мощности вычислительных средств (с); W — мощность вычислительных средств, нормированная на единичную номинальную мощность (например, на один процессор с тактовой частотой 1 ГГц) .

В случае биометрической идентификации потери, связанные с распараллеливанием вычислений, минимальны, поэтому при грубой оценке производительности данным фактором можно пренебречь.

Чтобы система справлялась с потоком заявок в течение суток, накладывается ограничение $T_{\text{сут}} < < 24$ ч. Резерв R системы (избыточность) определяется как

$$R = \frac{24 - T_{\text{сут}}}{24} = \\ = 1 - \frac{Nt_{\text{ср}}}{W}\,\frac{r_{\text{сут}}}{24} = 1 - \frac{Nt_{\text{ср}}}{W}\,r = 1 - tr\,,$$

где r — интенсивность потока заявок.

Избыточность в основном необходима в следующих случаях:

(1) сезонные колебания и резкие скачки нагрузки на биометрические серверы;

(2) выход из строя части вычислительных мощностей;

(3) плановый профилактический вывод из эксплуатации части вычислительных мощностей;

(4) сбой системы, приводящий к необходимости повторной обработки запросов.

В случае систем оперативной идентификации, где время ожидания ограничено минутами, помимо средней способности системы обеспечить обработку потока заявок требуется ограничить дисперсию времени ожидания, чтобы в моменты пиковой загрузки вычислительных мощностей проектное время ожидания не было превышено. В таком случае требуется проводить специальное исследование структуры потока заявок, чтобы определить максимально возможную интенсивность.

Как видно из формулы (1), основным фактором, ограничивающим производительность биометрической системы, является скорость сравнения биометрических образцов. Уменьшение времени сравнения положительно сказывается на производительности системы.

3 Алгоритмы интеграции

В случае одномодальной биометрической системы время сравнения и ошибки распознавания можно уменьшить только доработкой используемого специализированного биометрического программного обеспечения. Для мультибиометрических систем, в частности комбинации отпечатков пальцев и изображения лица, возможны несколько вариантов реализации биометрической идентификации, позволяющих корректировать эксплуатационные показатели.

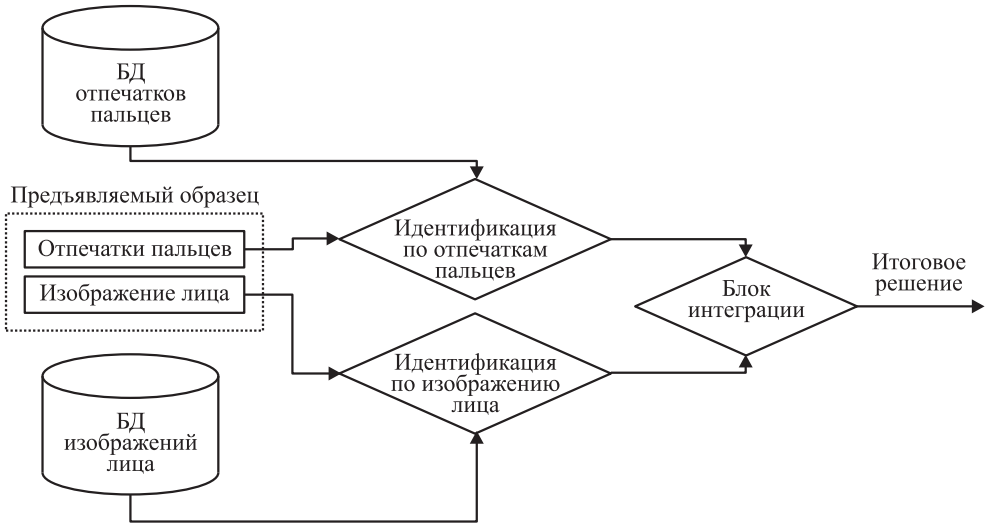
При реализации технологии одновременной идентификации по отпечаткам пальцев и изображению лица возможны две основные схемы интеграции:

(1) процессы сравнения отпечатков и изображения лица независимы (рис. 3, *а*);

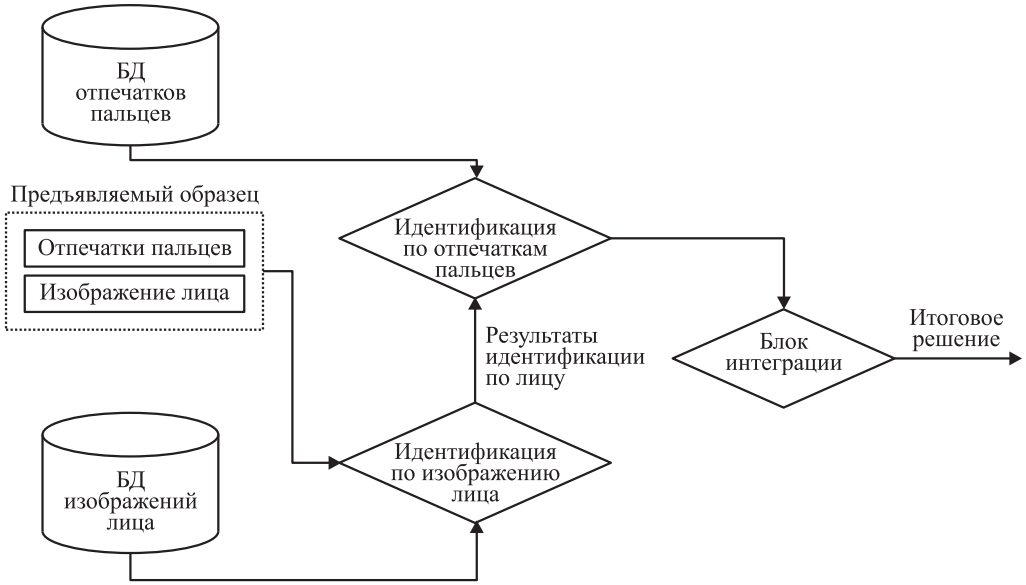
(2) процессы сравнения зависимы (рис. 3, *б*).

В [5–9] разработана методология интеграции биометрических систем в случае независимого сравнения. В таком случае достигаются минимальные возможные ошибки распознавания. Однако это приводит к потерям производительности. Во многих задачах, решаемых современными АДИС, качество идентификации в терминах ошибок распознавания является приемлемым. В то же время производительность остается достаточно низкой. Поэтому далее в статье мы сосредоточим основное внимание на схеме интеграции с зависимыми процессами идентификации, которая позволяет достичь прироста в производительности без потерь в качестве распознавания.

Рассмотрим реализацию идентификации более детально. На вход функции одномодальной биометрической идентификации поступают предъявляемая биометрическая запись или образец и биометрическая БД или линейный список биометрических записей. На выходе мы получаем меры сходства предъявляемой и хранимых в БД записей. На основе этой информации принимается решение, принадлежат ли записи одному человеку или нет. Большинство систем идентификации по отпечаткам пальцев и изображению лица используют пороговые методы принятия решения.



(*а*)



(*б*)

Рис. 3 Независимые (*а*) и зависимые (*б*) процессы идентификации

Увеличения производительности АДИС можно достичь, если по результатам идентификации по изображению лица принимать решения о целесообразности дальнейшего поиска по отпечаткам пальцев (схема реализации функции сравнения отпечатков пальцев и изображения лица приведена на рис. 4).

Как видно из рис. 4, в реализации функции сравнения биометрических образцов есть четыре терминальных состояния:

(1) 2 — при сравнении изображений лица принято решение об идентичности образцов (Accept), так как результат сравнения m_1 превышает определенный порог A_1;

(2) 4 — при сравнении изображений лица принято решение о различности образцов (Reject), так как мера сходства m_1 меньше некоторого минимального порога R_1;

(3) 5 — после сравнения отпечатков пальцев принято решение об идентичности образцов, суммарная мера сходства m_2 больше порога заданного порога A_2 (проблемы построения интегральной меры сходства при мультибиометрической идентификации изложены в [5, 6]);

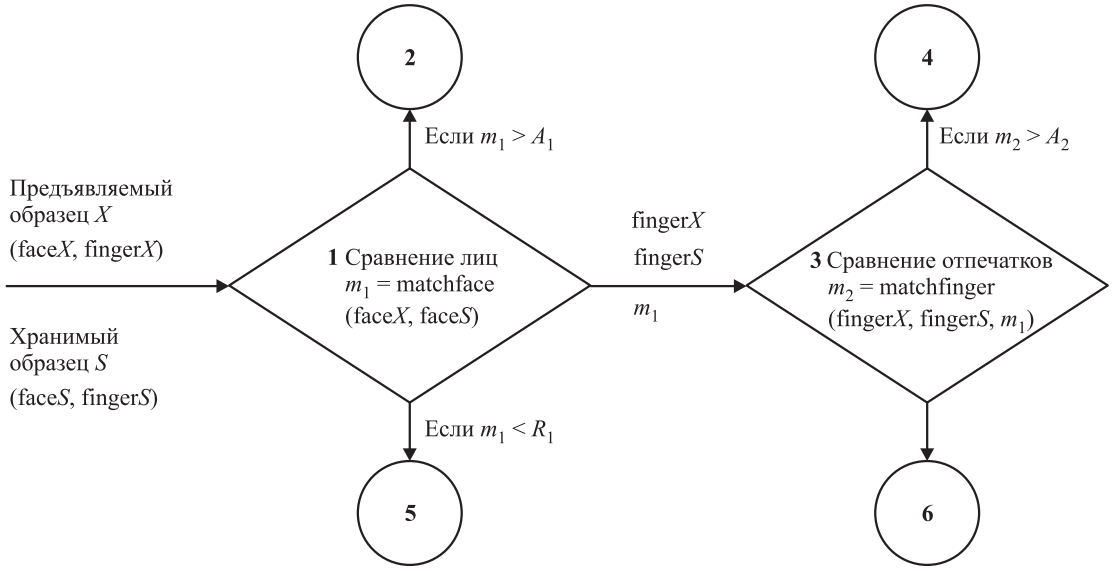


Рис. 4 Реализации функции мультибиометрического сравнения отпечатков пальцев и изображения лица

(4) 6 — по результатам сравнения принято решение о различности образцов.

Оценим статистические характеристики временных показателей выполнения функции мультибиометрического сравнения. Следует разделить следующие два случая:

(1) образцы принадлежат одному человеку (обозначим среднее время сравнения через t^g);

(2) образцы принадлежат разным людям (t^i).

В первом случае вероятности $m_1 \geq A_1$ и $m_1 < R_1$ являются стандартными показателями качества распознавания и обозначаются $\mathrm{TAR}(A_1)$, True Acceptance Rate, и $\mathrm{FRR}(R_1) = 1 - \mathrm{TAR}(R_1)$, ошибка первого рода. Соответственно среднее время сравнения «своих» будет складываться из двух слагаемых: время выполнения сравнения по изображению лица и времени сравнения по отпечаткам пальцев. Причем сравнение отпечатков пальцев будет проводиться только в случае $R_1 \leq m_1 < A_1$. Соответственно среднее суммарное время определяется по формуле:

$$t^g = t^g_{\mathrm{face}} + (1 - \mathrm{TAR}_{\mathrm{face}}(A_1) - \\ - \mathrm{FRR}_{\mathrm{face}}(R_1))\, t^g_{\mathrm{finger}} = \\ = t^g_{\mathrm{face}} + (\mathrm{FRR}_{\mathrm{face}}(A_1) - \mathrm{FRR}_{\mathrm{face}}(R_1))\, t^g_{\mathrm{finger}}, \quad (2)$$

где t^g_{face} — среднее время выполнения «своих» сравнений для изображения лица; t^g_{finger} — среднее время выполнения «своих» сравнений для отпечатков пальцев.

Во втором случае вероятности $m_1 \geq A_1$ и $m_1 < R_1$ выражаются аналогичным образом через ошибку второго рода $\mathrm{FAR}(A_1)$ и $\mathrm{TRR}(R_1) = 1 - \mathrm{FAR}(R_1)$, True Rejection Rate:

$$t^i = t^i_{\mathrm{face}} + (1 - \mathrm{FAR}_{\mathrm{face}}(A_1) - \\ - \mathrm{TRR}_{\mathrm{face}}(R_1))\, t^i_{\mathrm{finger}} = \\ = t^i_{\mathrm{face}} + (\mathrm{FAR}_{\mathrm{face}}(R_1) - \mathrm{FAR}_{\mathrm{face}}(A_1))\, t^i_{\mathrm{finger}}, \quad (3)$$

где t^i_{face} — среднее время выполнения «чужих» сравнений для изображения лица; t^i_{finger} — среднее время выполнения «чужих» сравнений для отпечатков пальцев.

4 Оценка производительности

Сосредоточим внимание на решении задачи оценки производительности в задаче массовой идентификации. Массовость сравнения позволяет при оценке производительности в значительной степени ориентироваться на формулы (2) и (3), так как при сравнении по большой базе входящие в формулу вероятности дают достаточно точную оценку времени идентификации.

При выполнении операции массовой идентификации доминируют операции сравнения «чужих». Так, если в базе зарегистрированы по одному образцу для N субъектов, то в процессе идентификации будет выполнено N «чужих» и одно «свое» сравнение. Соответственно при большом N и сравнимых по масштабу t^g и t^i доля времени t^g в процессе идентификации ничтожно мала, ей можно пренебречь. Поэтому при использовании одного вычислительного узла изменение $a(A_1, R_1)$ про-

изводительности АДИС за счет добавления идентификации по изображению лица с выбранными порогами A_1 и R_1 можно вычислить как отношение (3) к времени идентификации t^i_{finger} только по отпечаткам пальцев:

$$a(A_1,R_1)=\frac{t^i}{t^i_{\text{finger}}}=\frac{t^i_{\text{face}}}{t^i_{\text{finger}}}+ \\ +\left(\text{FAR}_{\text{face}}(R_1)-\text{FAR}_{\text{face}}(A_1)\right). \quad (4)$$

Во многих случаях для сравнения лица потребуется выделение отдельных вычислительных средств (обозначим их долю через g) за счет дактилоскопического вычислительного узла, в котором остается $h=1-g$ исходных вычислительных мощностей. Если пренебречь эффектом потерь распараллеливания (которые достаточно малы при биометрической идентификации), требуется отдельно вычислять скорости сравнения с учетом изменения вычислительной мощности. Обозначим среднее время биометрических сравнений на вычислительном узле единичной эталонной производительности через τ^i_{face} и τ^i_{finger} соответственно, тогда выражение (4) модифицируется следующим образом:

$$a(g,A_1,R_1)= \\ =\frac{\tau^i_{\text{face}}}{g\tau^i_{\text{finger}}}+\frac{\text{FAR}_{\text{face}}(R_1)-\text{FAR}_{\text{face}}(A_1)}{h}, \quad (5)$$

где $a(g,A_1,R_1)$ — изменение производительности с поправкой на перераспределение g вычислительных мощностей.

Учитывая реальные требования к современным системам биометрической идентификации, вероятность перехода в узел *2* (см. рис. 4) будет очень редким событием, поскольку на ошибку второго рода накладываются серьезные ограничения $\text{FAR}_{\text{face}}(A_1)<10^{-3}\div10^{-8}$, из которого определяется допустимое значение порога A_1. Соответственно в чужих сравнениях вероятность перехода практически нулевая и не влияет на среднее время. Поэтому следует отметить, что в большинстве приложений при оптимизации производительности придется отказаться от возможности принятия положительного решения на основе идентификации только по изображению лица, так как прирост минимальный, а риски неправильной оценки FAR_{face} — высоки. В то же время порог R_1 будет регулироваться другим ограничением: максимально допустимым уровнем ошибки 1-го рода, $\text{FRR}_{\text{face}}(R_1)\approx0$.

При определении распределения вычислительных мощностей наибольший прирост производительности наблюдается в точке g_{opt} минимума (5), которая при фиксированных A_1 и R_1 может быть найдена из соотношения

$$\frac{\partial a}{\partial g}(g_{\text{opt}})=0. \quad (6)$$

Преобразуя (6), получаем следующее уравнение для g_{opt}:

$$\frac{\partial a}{\partial g}(g_{\text{opt}})=-\frac{\tau^i_{\text{face}}}{g^2_{\text{opt}}\tau^i_{\text{finger}}}+ \\ +\frac{\text{FAR}_{\text{face}}(R_1)-\text{FAR}_{\text{face}}(A_1)}{h^2_{\text{opt}}}=0.$$

Соответственно получаем следующее условие на g_{opt}:

$$\frac{\tau^i_{\text{finger}}\left(\text{FAR}_{\text{face}}(R_1)-\text{FAR}_{\text{face}}(A_1)\right)}{\tau^i_{\text{face}}}= \\ =\frac{(1-g_{\text{opt}})^2}{g^2_{\text{opt}}}=\left(\frac{1}{g_{\text{opt}}}-1\right)^2. \quad (7)$$

Так как g_{opt} является долей вычислительных средств, выделенных на идентификацию лица, значение g_{opt} находится в интервале от 0 до 1. Поэтому из (7) находим единственное решение:

$$\frac{1}{g_{\text{opt}}}= \\ =1+\sqrt{\frac{\tau^i_{\text{finger}}(\text{FAR}_{\text{face}}(R_1)-\text{FAR}_{\text{face}}(A_1))}{t^i_{\text{face}}}}; \quad (8)$$

$$\sqrt{\frac{\tau^i_{\text{finger}}(\text{FAR}_{\text{face}}(R_1)-\text{FAR}_{\text{face}}(A_1))}{t^i_{\text{face}}}}=\frac{h_{\text{opt}}}{g_{\text{opt}}}. \quad (9)$$

Так как на краях интервала

$$\lim_{g\to0}a(g,A_1,R_1)=+\infty$$

и

$$\lim_{g\to1}a(g,A_1,R_1)=+\infty,$$

единственная точка экстремума является искомым минимумом. Подставляя (8) и (9) в (5), получаем следующее минимальное значение функции a:

$$a(g_{\text{opt}},A_1,R_1)=\frac{\tau^i_{\text{face}}}{g_{\text{opt}}t^i_{\text{finger}}}+ \\ +\frac{\sqrt{\text{FAR}_{\text{face}}(R_1)-\text{FAR}_{\text{face}}(A_1)}}{g_{\text{opt}}}\sqrt{\frac{\tau^i_{\text{face}}}{\tau^i_{\text{finger}}}}=\left(1+ \right. \\ \left.+\sqrt{\frac{\tau^i_{\text{finger}}(\text{FAR}_{\text{face}}(R_1)-\text{FAR}_{\text{face}}(A_1))}{\tau^i_{\text{face}}}}\right)\left(\frac{\tau^i_{\text{face}}}{\tau^i_{\text{finger}}}+ \right. \\ \left.+\sqrt{\text{FAR}_{\text{face}}(R_1)-\text{FAR}_{\text{face}}(A_1)}\sqrt{\frac{\tau^i_{\text{face}}}{\tau^i_{\text{finger}}}}\right).$$

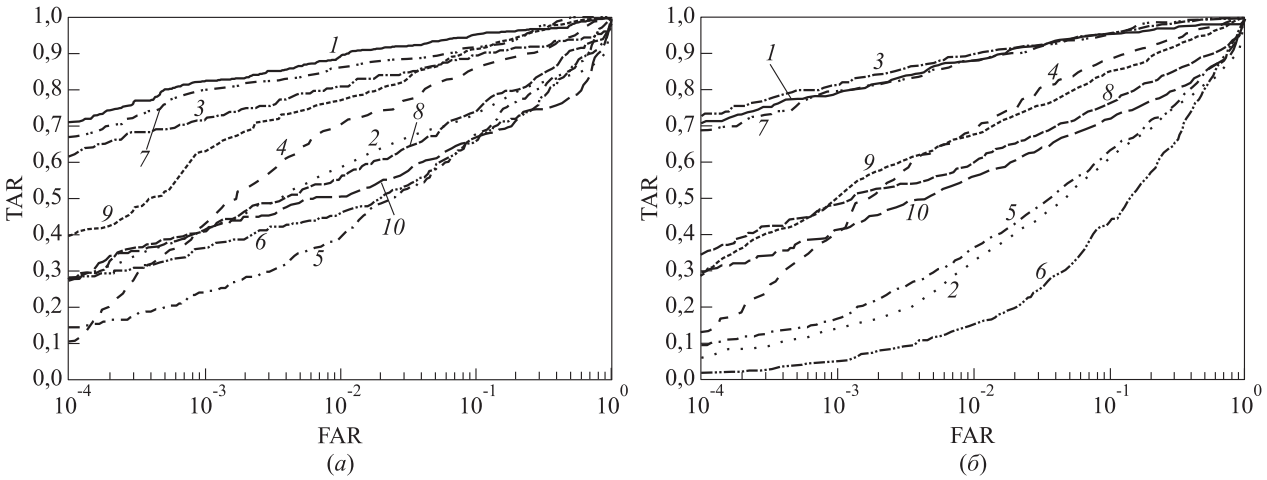


Рис. 5 Ошибки распознавания по изображению лица (внутри помещения, съем биометрии в разные дни) по данным [10]: (*а*) Fig. L.18, стандартное разрешение; (*б*) Fig. L.48, высокое разрешение. Производители систем распознавания по изображению лица: *1* — Eyematic; *2* — AcSys; *3* — Cognitec; *4* — C-VIS; *5* — DreamMIRH; *6* — Iconquest; *7* — Identix; *8* — Imagis; *9* — Visage; *10* — VisionSphere

Обозначая

$$c^2 = \frac{\tau^i_{\text{face}}}{\tau^i_{\text{finger}}}\,;$$

$$d^2(A_1, R_1) = \text{FAR}_{\text{face}}(R_1) - \text{FAR}_{\text{face}}(A_1)$$

получаем следующее выражение для оптимального значения изменения производительности:

$$a(g_{\text{opt}}, A_1, R_1) = \left(1 + \frac{d}{c}\right)\left(c^2 + dc\right) = (c+d)^2\,. \quad (10)$$

5 Моделирование

Для оценки возможного прироста производительности исследуем статистику тестирований биометрических технологий NIST FRVT2006, NIST FRVT2002 (лицевая биометрия) и NIST FpVTE (отпечатки пальцев) [10–12]. Согласно протоколам тестирований, производительность современных систем распознавания фронтальных лиц колеблется в интервале от 50 000 до 330 000 сравнений в секунду в расчете на одноядерный процессор тактовой частоты 2,5 ГГц. Аналогичная характеристика для отпечатков пальцев составляет примерно 5000 сравнений в секунду. Соответственно на основании этих данных мы имеем следующую оценку для параметра c изменения производительности (10):

$$c^2 = \frac{\tau^i_{\text{face}}}{\tau^i_{\text{finger}}} = 0{,}03 \div 0{,}1\,;$$

$$c = 0{,}123 \div 0{,}316\,.$$

Для определения возможных порогов A_1 и R_1, влияющих на оценку параметра d, проанализируем динамику зависимости FAR и FRR (рис. 5) от порога. Напомним, что порог R_1 определяется из условия $\text{FRR}_{\text{face}}(R_1) \approx 0$, порог A_1 — из условия $\text{FAR}_{\text{face}}(A_1) < 10^{-3} \div 10^{-8}$ соответственно.

На основании приведенных графиков получаем, что предложенная схема ускорения дает следующее улучшение производительности:

$$d^2 = \text{FAR}_{\text{face}}(R_1) - \text{FAR}_{\text{face}}(A_1) = 0{,}05 \div 0{,}2\,;$$

$$d = 0{,}223 \div 0{,}447\,;$$

$$c + d = 0{,}346 \div 0{,}763\,;$$

$$(c+d)^2 = 0{,}119 \div 0{,}582\,.$$

Таким образом, получаем, что среднее время сравнения интегрированной идентификационной системы составляет $0{,}119 \div 0{,}582$ от АДИС, построенной на вычислительных средствах той же производительности. Соответственно скорость идентификации возрастает в 2–10 раз в зависимости от качества используемой лицевой биометрии.

С точки зрения перспективы соотношение $c = 0{,}123 \div 0{,}316$ вряд ли улучшится, при этом прирост качества автоматического распознавания лиц может довести d^2 до уровня в 1%, что дает предел ускорения примерно в 20 раз.

6 Заключение

Предложенный метод интеграции АДИС и лицевой биометрии позволяет значительно увеличить скорость мультибиометрической идентификации. В качестве преимуществ разработанных методов отметим следующие:

- при современных технологиях распознавания разработанная информационная технология позволяет ускорить идентификацию в 2–10 раз;
- алгоритмы интеграции не зависят от специфики используемых технологий распознавания по отпечаткам пальцев и лицу;
- алгоритмы интеграции могут быть реализованы на «верхнем» уровне биометрического API (Application Program Interface — прикладной программный интерфейс).

В качестве направления дальнейших работ можно выделить обобщение разработанных подходов на общий случай мультибиометрической идентификации.

Литература

1. *Ushmaev O. S., Novikov S. O.* Integral criteria for large-scale multiple fingerprint solutions // Biometric technology for human identification / Eds. A. K. Jain, N. K. Ratha. SPIE Proceedings, 2004. Vol. 5404. P. 534–543.
2. *Dizard III, Wilson P.* FBI plans major database upgrade // Government Computer News. http://www.gcn.com/print/25_26/41792-1.html?page=1.
3. *Болл Р. М., Коннел Дж. Х., Панканти Ш., Ратха Н. К., Сеньор Э. У.* Руководство по биометрии. — М.: Техносфера, 2007.
4. *Ушмаев О. С., Босов А. В.* Реализация концепции многофакторной биометрической идентификации в интегрированных аналитических системах // Бизнес и безопасность в России, 2008. № 49. С. 104–105.
5. *Ушмаев О. С., Синицын И. Н.* Опыт проектирования многофакторных биометрических систем // Тр. VIII международной научно-технической конференции «Кибернетика и высокие технологии XXI века». Т. 1. С. 17–28.
6. *Синицын И. Н., Новиков С. О., Ушмаев О. С.* Развитие технологий интеграции биометрической информации // Системы и средства информатики, 2004. Вып. 14. С. 5–36.
7. *Ushmaev O., Novikov S.* Biometric fusion: Robust Approach // MMUA 06 Proceedings. Toulouse, France, May 11–12, 2006.
8. *Ушмаев О. С., Босов А. В.* Реализация концепции многофакторной биометрической идентификации в интегрированных аналитических системах // Системы высокой доступности, 2007. 4, Т. 3. С. 13–23.
9. *Ушмаев О. С., Синицын И. Н.* Программная реализация мультибиометрической идентификации в интегрированных аналитических приложениях // Тр. IX международной научно-технической конференции «Кибернетика и высокие технологии XXI века». Воронеж, 13–15 мая 2008. Т. 2. С. 735–746.
10. NIST FRVT2002. Evaluation Report. http://www.frvt.org/FRVT2002/.
11. NIST FRVT2006. Evaluation Report. http://www.frvt.org/FRVT2006/.
12. NIST FpVTE. Evaluation Report. http://fpvte.nist.gov.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2008. Т. 2. Вып. 4. С. 74–84

РЕГИОНЫ ВРЕМЕНИ КАК ОБЪЕКТЫ ОПЕРАЦИОННОЙ СИСТЕМЫ ОБЩЕГО НАЗНАЧЕНИЯ

В. Ю. Егоров[1], Е. А. Матвеев[2]

Аннотация: В статье предлагается ввести в состав операционных систем общего назначения новый тип объекта — «регион времени», который позволяет придать им свойства операционных систем реального времени.

Ключевые слова: операционная система; реальное время; регион времени; расширение языка Си; аппаратный контроллер прерываний; многопроцессорная система

1 Введение

В настоящее время все операционные системы принято делить на два больших класса: операционные системы общего назначения (называемые также универсальными) и операционные системы специального назначения. В состав класса операционных систем специального назначения отдельным пунктом входят операционные системы реального времени. В данной статье предлагается подход, позволяющий придать универсальной операционной системе черты системы реального времени.

Реализация подсистемы реального времени в ядре операционной системы общего назначения возможна, если в состав типов объектов операционной системы ввести особый тип объекта: «регион времени». Данный тип объекта характеризует участок кода программы, который должен быть выполнен за определенное время.

Регионы времени задаются границами, определяющими действие региона. Начальные границы регионов во времени и в коде совпадают. Реакция системы на исполнение региона времени зависит от того, какая из конечных границ региона будет достигнута раньше: граница во времени или граница в коде. Если первой будет достигнута конечная граница региона в коде, то выполнение последующего кода будет продолжено без какой-либо реакции от подсистемы реального времени. Если первой будет достигнута конечная граница региона во времени, то подсистема реального времени прерывает нормальную последовательность выполнения команд и управление передается в ветвь, отвечающую за обработку состояния истечения времени (тайм-аута).

Регионы времени обладают следующими характеристиками:

- начало региона во времени;
- конец региона во времени;
- начало региона в коде системы;
- конец региона в коде системы.

В качестве временны́х характеристик региона времени может быть выбрано либо абсолютное, либо относительное время. Относительное время считается от текущей точки во времени. Относительное время можно также привязать ко времени работы текущего процесса (нити) либо ко времени работы всей системы. Характеристики региона времени иллюстрируются на рис. 1.

Изменение порядка следования команд при достижении конечной границы региона во времени характеризуется двумя параметрами:

(1) временем, в течение которого система определяет, что произошел тайм-аут, так называемое «время отсечки»;

(2) временем, которое система тратит на изменение порядка следования команд, так называемое «время переключения».

После того как исполнение кода программы вошло в регион времени, диспетчер задач операционной системы общего назначения начинает работать как диспетчер задач операционной системы реального времени. Реализация объектов регионов времени требует изменения в подходах к проектированию большого числа компонентов ядра операционной системы.

[1]Пензенский государственный университет; ООО Научно-техническое предприятие «Криптософт», vec@cryptosoft.ru
[2]ООО Научно-техническое предприятие «Криптософт», eugene@cryptosoft.ru

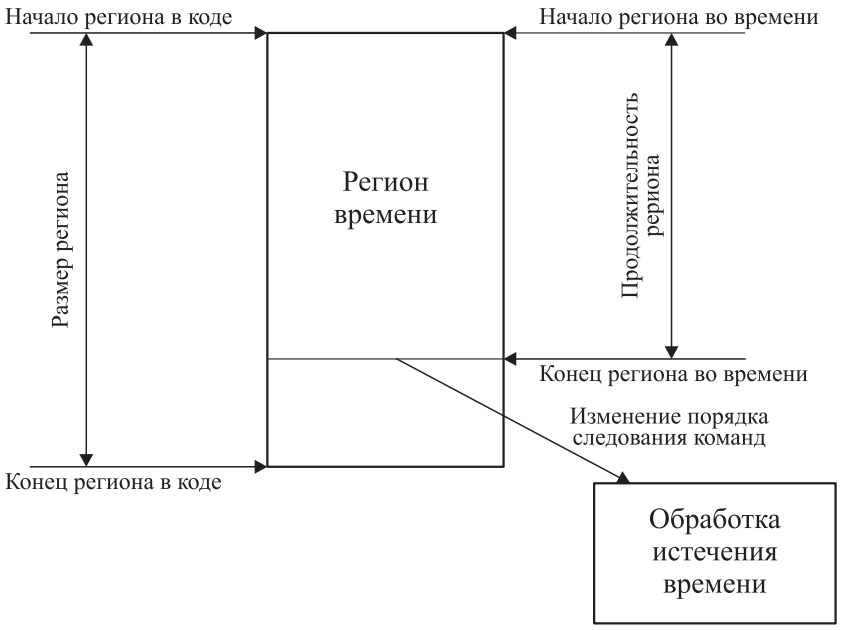


Рис. 1 Понятия регионов времени

Авторам исследования не удалось обнаружить в литературе и сети Интернет упоминаний о представлении регионов времени в виде объектов операционной системы общего назначения.

Одной из особенностей высокоточного таймера является использование его в режиме однократного программирования — на каждый регион времени таймер программируется заново. Программирование высокоточного таймера на периодическое срабатывание при программировании регионов времени просто не требуется. В отличие от высокоточного таймера стандартный системный таймер должен функционировать в периодическом режиме, поскольку на него возлагается обязанность актуализации системного времени.

Длительность времени переключения при возникновении тайм-аута зависит от характеристик операционной системы. Одной из важнейших характеристик является понятие аппаратного приоритета — IRQL (Interrupt Request Level) [1]. IRQL представляет собой целое беззнаковое число в интервале от 0 (PASSIVE_LEVEL) до HIGH_LEVEL. В зависимости от вида операционной системы HIGH_LEVEL может различаться. Однако аппаратные прерывания могут происходить только до значения MAX_DEVICE. Это значение определяется контроллером прерываний и равно 15.

Современные аппаратные средства в составе компьютера предоставляют возможность аппаратной поддержки для реализации диспетчера задач операционной системы. В основном это касается механизмов обслуживания прерываний. Все современные компьютеры, работающие на платформе IA-32, имеют в своем составе расширенный контроллер прерываний — APIC (Advanced Programmable Interrupt Controller) [2].

Контроллер APIC состоит из двух частей (рис. 2). Одна — в составе центрального процессора, называемая LocalApic [4, 3], вторая — в составе одной из микросхем материнской платы компьютера, называемая IoApic [2, 5]. И LocalApic, и IoApic могут присутствовать в системе в нескольких экземплярах [6]. Каждый IoApic имеет в своем составе некоторое число линий (обычно 24), используемых как входы прерываний аппаратных устройств. Если объединить несколько IoApic, то можно значительно увеличить число входов. В современных серверах имеется более пятидесяти линий для подключения

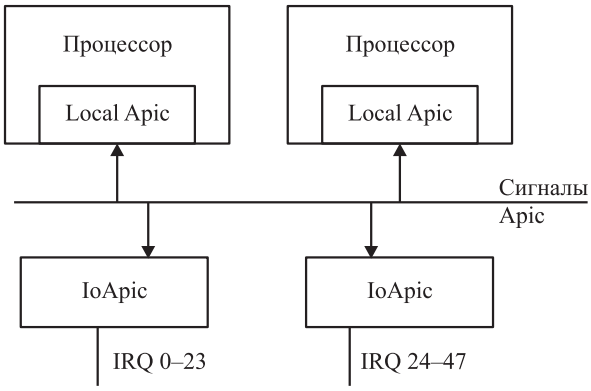


Рис. 2 Организация расширенного контроллера прерываний

сигналов прерываний от аппаратных устройств. Таким способом достигается расширение числа аппаратных прерываний от подключаемых устройств.

Отличительной особенностью контроллера является то обстоятельство, что понятие IRQL в нем введено аппаратно. Такая организация позволяет обрабатывать прерывания значительно быстрее. При возникновении сигнала прерывания контроллер APIC переключает процессор на обработку прерывания и аппаратно повышает IRQL до уровня, заданного при установке обработчика на прерывание. После этого могут возникнуть только прерывания, уровень IRQL которых выше, чем текущий. Те сигналы прерываний, чей IRQL ниже или равен текущему, остаются ожидать понижения IRQL, выполняемого операционной системой по завершении обработчика прерывания.

Благодаря APIC сводится к минимуму скорость реакции системы на возникновение прерываний. Это имеет очень большое значение при реализации регионов времени, так как уменьшает время переключения задачи на обработчик истечения времени.

2 Зависимость регионов времени от аппаратного приоритета

Необходимо обратить особое внимание на использование регионов времени при различных уровнях аппаратного приоритета (IRQL). По результатам проведения исследований было выяснено, что функционирование регионов времени возможно на любых IRQL вплоть до MAX_DEVICE, т. е. до максимального уровня аппаратного прерывания компьютера. Однако для этого, во-первых, необходима дополнительная поддержка со стороны операционной системы, а во-вторых, при работе с регионами времени требуется соблюдать следующее правило: уровень аппаратного приоритета не должен понижаться в течение выполнения всего региона времени. Несоблюдение правила должно отслеживаться и непременно приводить к особой системной ошибке: «понижение IRQL в течение региона времени». Работа операционной системы после возникновения такой ошибки должна прекращаться. Принцип функционирования региона времени на высоких уровнях IRQL иллюстрируется на рис. 3.

Если допустить возможность понижения IRQL в процессе выполнения региона времени, то диспетчер задач операционной системы может переключить контекст нити. При этом, во-первых, возрастет длительность выполнения региона времени, а во-вторых, и как следствие первого, может возникнуть тайм-аут. Возникновение тайм-аута приведет к обработке истечения времени, которая может быть произведена уже в контексте другой нити, что недопустимо.

Алгоритм действий, выполняемых операционной системой при старте региона времени, следующий:

1. Запомнить адрес, куда будет передаваться управление после возникновения и обработки тайм-аута.
2. Запомнить текущий уровень IRQL.
3. Вычислить абсолютное время возникновения ситуации тайм-аута.
4. Запрограммировать таймер на ближайшее по времени срабатывание тайм-аута для списка всех текущих регионов времени.

При возникновении тайм-аута алгоритм действий будет следующим:

1. Получить значение текущего IRQL.
2. Если текущий IRQL равен IRQL региона времени и он выше или равен DISPATCH_LEVEL, то сразу перейти к обработчику истечения времени.
3. Если текущий IRQL равен IRQL региона времени и он ниже DISPATCH_LEVEL, то произвести переключение контекста на нить региона времени и затем перейти к обработчику истечения времени.
4. Если текущий IRQL выше, чем IRQL региона времени, то выставить особое прерывание в контроллере прерываний с IRQL, равным IRQL региона времени, по которому управление получит обработчик истечения времени.

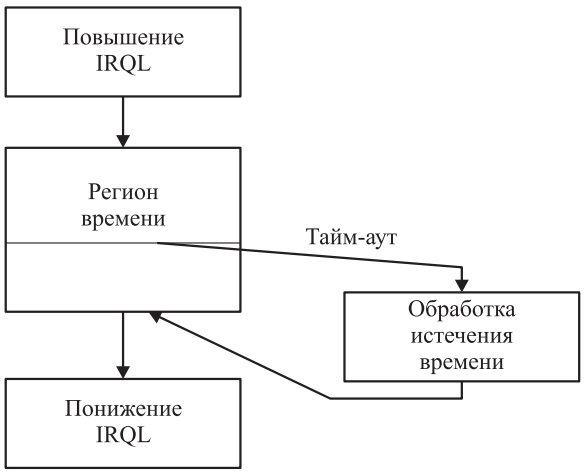


Рис. 3 Использование регионов времени на высоких IRQL

Отсечка по тайм-ауту должна при этом всегда срабатывать на максимально высоком уровне аппаратного приоритета, т. е. вне зависимости от наличия ждущих обработки аппаратных прерываний.

При использовании представленного выше алгоритма работы с регионами времени можно регулировать «жесткость» реального времени системы. Таким образом, работа с регионом времени на PASSIVE_LEVEL будет, по сути, представлять собой улучшенный вариант программируемого таймера. Работа с регионом времени на APC_LEVEL и DISPATCH_LEVEL будет соответствовать системам мягкого реального времени, т. е. таким системам, где допускается возникновение тайм-аута. Работа с регионом времени на MAX_DEVICE будет соответствовать жесткому реальному времени такому, которое обычно используется в специализированных операционных системах. Работа региона времени на уровнях IRQL, промежуточных между PASSIVE_LEVEL и MAX_DEVICE, будет представлять собой градации характеристики реального времени от более мягкого к более жесткому.

Длительность времени переключения системы на обработчик истечения времени при IRQL = PASSIVE_LEVEL определяется общей загруженностью процессора (процессоров) системы. Обработчик истечения времени получит управление после того, как закончат работу все прерывания, возникшие в текущий момент времени, а также нити, имеющие более высокий приоритет.

Длительность времени переключения системы на обработчик истечения времени при IRQL = MAX_DEVICE будет равна длительности времени отсечки плюс несколько десятков тактов процессора на передачу управления обработчику истечения времени.

3 Вложенность регионов времени

Абстракция регионов времени вполне допускает возможность вложенности одного региона времени в другой и даже их пересечения. Вложенность регионов времени представляет собой вложенность как участков кода, так и интервалов времени одного региона в другой. Пересечение регионов представляет собой ситуацию, когда один регион времени стартует раньше другого, но и завершается раньше во времени. Если при этом конечная точка кода первого региона также находится до конечной точки второго региона, то такую ситуацию будем называть *перекрестком регионов*. Сказанное иллюстрируется на рис. 4.

Реализация перекрестка регионов на практике вполне осуществима. Однако если в данном случае

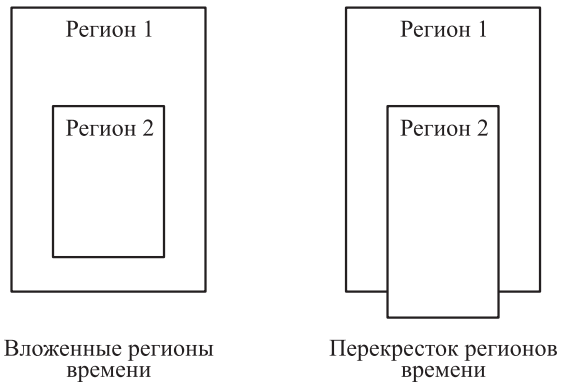


Рис. 4 Иллюстрация одновременного действия двух регионов времени

возникает срабатывание обоих регионов времени по тайм-ауту, то перекрестный участок кода будет выполнен как минимум дважды: один раз при срабатывании по тайм-ауту первого региона во времени, а второй раз при срабатывании по тайм-ауту второго региона времени. В худшем случае в коде перекрестка возникнет временная петля (часто описываемая фантастами), когда процессор будет вынужден бесконечно выполнять один и тот же участок кода. Поэтому от практической реализации перекрестка регионов времени было решено отказаться.

Ниже описана реализация регионов времени с использованием расширений языка программирования Си, которая полностью исключает возможность создания перекрестка регионов времени, но при этом допускает возможность их вложения.

4 Организация списка регионов времени

В процессе работы операционной системы вполне возможна ситуация, при которой своей очереди на срабатывание дожидаются одновременно несколько регионов времени. Необходимо рассмотреть вопрос, как именно должна быть организована в памяти последовательность регионов времени. Решение данной задачи возможно следующими способами:

(1) привязка регионов времени к нити, для которой они предназначены;

(2) привязка регионов времени к системному таймеру.

В первом способе при возникновении прерывания таймера следует просмотреть все нити, содержащие в себе регионы времени, и в случае истече-

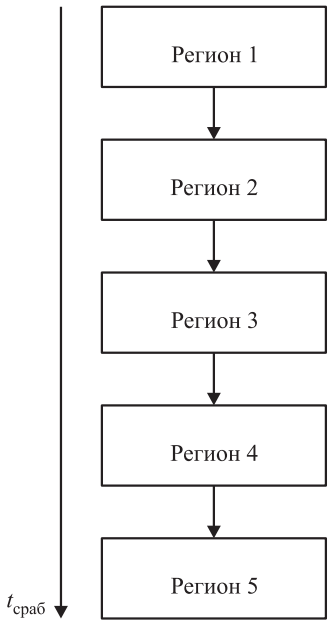


Рис. 5 Организация списка регионов времени

ния тайм-аута произвести отсечку региона времени. Во втором способе формируется единый список регионов времени. Этот список должен быть отсортирован по времени срабатывания. При возникновении прерывания таймера система начинает последовательный просмотр регионов времени с самого раннего времени срабатывания. Для всех сработавших регионов времени производится отсечка. Как только в списке встречается еще не сработавший регион времени, просмотр заканчивается. Данный алгоритм иллюстрирует рис. 5.

Первый способ организации регионов времени не требует предварительной сортировки, и потому предпочтительнее, когда число регионов времени невелико. В случае большого количества регионов времени более предпочтительным следует признать второй способ. При создании региона времени сортировка в этом случае сводится к поиску места для вновь созданного региона. При срабатывании таймера отсутствие необходимости просмотра всего списка даст весомый выигрыш во времени.

5 Уничтожение погрешности измерения времени во вложенных регионах времени

Рассмотрим ситуацию, когда в одном длительном регионе времени (назовем его регионом 1) исполняется цикл. В каждой итерации цикла заводится другой регион времени малой длительности, который назовем регионом времени 2, так, как это показано ниже.

```
⟨Старт региона 1 в коде программы⟩
for (dwIndex=0; dwIndex < 4; ++dwIndex)
{
    ⟨Старт региона 2 в коде программы⟩
    . . .
    ⟨Конец региона 2 в коде программы⟩
}
⟨Конец региона 1 в коде программы⟩
```

В результате выполнения программы во времени образуется последовательность регионов времени, показанная на рис. 6.

Для программирования регионов времени используется высокоточный таймер в составе LocalApic процессора. Как уже отмечалось, высокоточный таймер в составе процессора один, и его каждый раз требуется программировать однократно для каждого вложенного региона времени. Это создает ситуацию, когда таймер должен будет программироваться на регион времени 2 в начале каждой итерации цикла и на регион времени 1 в конце итерации. Таким образом, на регион времени 1 таймер будет программироваться 5 раз. Всего число программирований таймера на регион времени 1 зависит от числа итераций цикла:

$$\langle\text{число программирований региона 1}\rangle = \\ = \langle\text{число итераций}\rangle + 1\,.$$

Иллюстрация числа программирований таймера приведена на рис. 7.

При задании региона времени для программиста удобнее всего задавать его длительность. Программирование таймера также требует задания интервала времени. Однако, как показывает вышеприведенный пример, если каждый раз при программировании таймера на регион времени 1 вычислять оставшуюся длительность региона путем вычитания уже прошедшего времени, то в результате для региона 1 будет накапливаться систематическая погрешность вычисления оставшегося времени. Каждое программирование таймера также занимает определенный интервал времени, поэтому при большом числе итераций цикла систематическая погрешность сильно возрастает.

Выход из данной ситуации заключается в вычислении не длительности региона времени, а конечной точки во времени, в которой регион времени должен завершиться. Эта конечная точка во времени должна быть вычислена уже на этапе старта

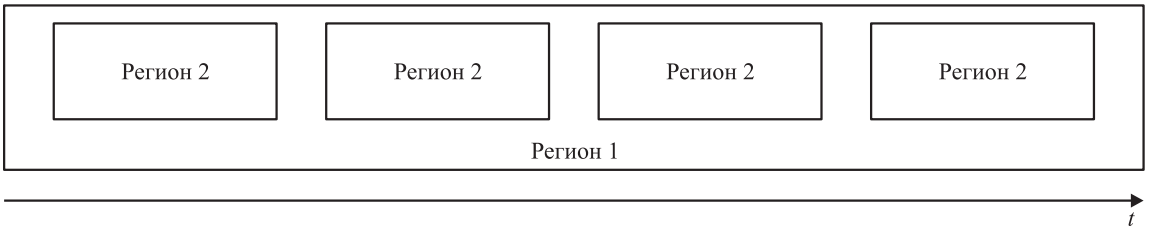


Рис. 6 Вложенность регионов времени в составе цикла

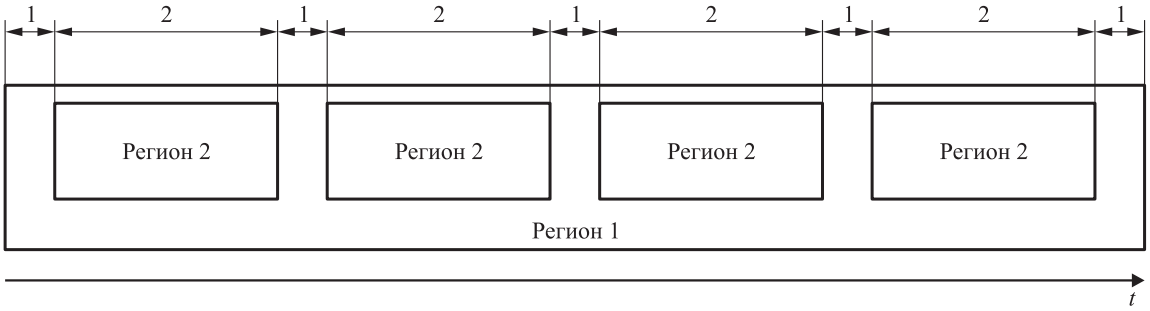


Рис. 7 Последовательность программирования таймера на регионы времени

региона времени. При каждом программировании таймера необходимо получить разность между временем истечения региона и текущим временем. Эта разность и будет определять время, на которое следует программировать таймер LocalApic.

Однако данный способ также не лишен недостатков. Системное время в операционной системе вычисляется с помощью стандартного таймера компьютера (аналога микросхемы 8253), который, как известно, имеет весьма низкую точность. Тактовая частота системного таймера равна 1,193181 МГц [7]. Это означает, что погрешность измерения времени таймером составляет 800 нс. Это уже достаточно большая величина, чтобы нивелировать все преимущества высокоточного таймера LocalApic. При этом в современных операционных системах счетчик времени инкрементируется не с приведенной частотой, а по возникновению прерывания. В ОС Windows, например, такое прерывание возникает примерно каждые 10 мс, и погрешность системного времени также составляет 10 мс [8]. Конечно, такая точность измерения времени неприемлема для подсистемы реального времени.

Выход из ситуации возможен с использованием счетчика тактов процессора. Такой счетчик (Time Stamp Counter — TSC) содержится в каждом процессоре, начиная с процессоров Intel Pentium. Расширенный контроллер прерываний в составе процессоров появился позже, следовательно, в каждом процессоре, где есть расширенный контроллер прерываний, имеется также и счетчик тактов.

Счетчик тактов представляет собой 64-битный регистр, значение которого монотонно возрастает с каждым тактом частоты процессора. Для процессора с тактовой частотой 2 ГГц погрешность измерения времени счетчиком тактов составляет 0,5 нс. Такой точности измерения времени более чем достаточно даже для систем жесткого реального времени.

Для того чтобы таймер LocalApic и счетчик тактов процессора можно было использовать для измерения времени, необходимо произвести их калибровку. Единственным источником в компьютере с известной тактовой частотой является стандартный системный таймер. С его помощью необходимо производить калибровку частоты процессора и частоты системной шины при инициализации операционной системы.

При старте региона времени необходимо получить текущее значение счетчика тактовой частоты, перевести длительность интервала времени в число тактов процессора и вычислить значение счетчика тактов процессора при завершении региона времени. Это значение и будет храниться в системе в течение действия региона времени. Алгоритм программирования таймера LocalApic в данном случае будет выглядеть следующим образом:

1. Получить текущее значение счетчика тактов процессора.

2. Вычислить разницу между конечным значением счетчика тактов и его текущим значением.

3. Выполнить перевод количества тактов процессора в количество тактов таймера LocalApic.
4. Запрограммировать таймер LocalApic.

Приведенный алгоритм позволяет уменьшить погрешность измерения времени до погрешности таймера LocalApic. При частоте системной шины 200 МГц погрешность составит всего 5 нс. Такой способ также полностью исключает систематическую погрешность при наличии вложенных регионов времени.

6 Реализация регионов времени на многопроцессорной системе

При наличии в составе аппаратной платформы нескольких процессоров или процессорных ядер задача управления регионами времени усложняется. В данной статье будет рассматриваться только случай симметричной многопроцессорной схемы SMP (Symmetric Multiprocessor) с однородным доступом к памяти UMA (Uniform Memory Access) [9]. Такая структура используется в настоящий момент в многоядерных и многопроцессорных вычислительных системах от компаний Intel и AMD.

Наличие многопроцессорной архитектуры предоставляет дополнительные преимущества, но и вносит дополнительные сложности в реализацию регионов времени. С одной стороны, наличие нескольких процессоров позволяет одновременно выполнять код нескольких регионов времени. С другой стороны, при использовании нескольких процессоров в системе возникают специфические задержки, связанные с синхронизацией процессоров.

В SMP-системах все процессоры имеют равные права на доступ к памяти. Память системы общая для всех процессоров, и изменение одним из процессоров одной ячейки памяти приводит к ее изменению на всех процессорах в системе. Данный механизм реализован аппаратно. Однако в процессорах имеются данные, которые не обновляются аппаратно. К ним относится содержимое регистра CR3, который хранит адрес таблицы страниц текущего процесса. Перезаписывание регистра CR3 приводит к сбросу буферов TLB (Translation Lookaside Buffer — буфер быстрого преобразования адреса) процессора. Если два процессора в системе выполняют код одного и того же процесса, то при изменении таблицы страниц в одном из процессоров необходимо также изменять таблицу страниц другого процессора в системе.

Для выполнения данной операции процессоры в многопроцессорной системе время от времени приходится синхронизировать. На практике это выражается в том, что по команде от одного из процессоров нормальная последовательность выполнения команд прерывается и процессор начинает выполнять действия по синхронизации.

Несмотря на то что перезаписывание регистра CR3 выполняется всего за две процессорные инструкции (считывание регистра CR3 в один из регистров общего назначения и повторная запись этого же значения в регистр CR3), реально эта операция выполняется в течение нескольких сотен тактов процессора. Данная операция выполняется так часто, как скоро возникают изменения в таблицах страниц. Откладывать ее выполнение нельзя, поскольку иначе у разных процессоров в системе возникнут различия в представлении адресных пространств. Операция синхронизации процессоров должна выполняться вне зависимости от наличия или отсутствия аппаратных прерываний в системе. Поэтому логичнее всего команду синхронизации и обработчик установить на немаскируемое прерывание (NMI, nonmaskable interrupt). Вход процессора в обработчик прерывания и выход из него вносят дополнительные задержки в работу процессора.

Несомненно, синхронизация процессоров в многопроцессорной системе снижает точность выполнения регионов времени. Это является основным недостатком многопроцессорной системы.

С другой стороны, SMP-системы обладают также неоспоримыми преимуществами. Наличие в каждом из процессоров (процессорных ядер) собственного высокоточного таймера позволяет одновременно обрабатывать несколько регионов времени в разных процессах, что повышает гибкость системы в целом.

Реализация регионов времени в многопроцессорной системе требует изменения алгоритма работы по программированию таймера. Теперь требуется обеспечить балансировку загрузки процессоров регионами времени. Алгоритмы балансировки могут быть достаточно сложны и разнообразны, но при этом они будут дополнительно занимать процессорное время, поэтому предпочтительным представляется наиболее простой алгоритм.

Регион времени всегда привязан к нити, на которой он выполняется. В многопроцессорной системе всегда реализован тот или иной принцип распределения нитей между процессорами. Поэтому наиболее оптимальным представляется принцип, по которому ответственным за регион времени назначается тот процессор, на котором был обработан старт региона времени. В результате происходит привязка региона времени к процессору и дополнительного алгоритма балансировки регионов времени между процессорами не требует-

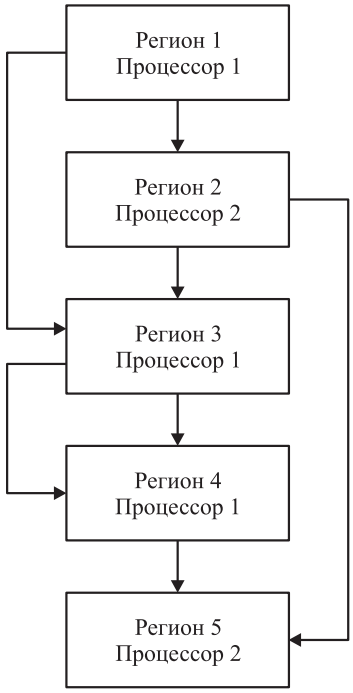


Рис. 8 Привязка регионов времени к процессорам

ся. Для каждого процессора в системе становится необходимым формировать список регионов времени, привязанных к данному процессору. Данная схема иллюстрируется на рис. 8.

Когда какой-либо из регионов времени срабатывает, процессор фиксирует отсечку региона времени, изменяет текущий контекст нити и ставит ее на обработку. Затем процессор выбирает из списка следующий регион времени, отнесенный к данному процессору, и программирует свой таймер LocalApic заново.

7 Реализация регионов времени с помощью функций

Любая реализация регионов времени требует возможности запоминания точки, в которую будет передано управление при завершении региона времени с тайм-аутом. Самый простой способ реализации данного механизма заключается в использовании точки старта региона времени также и в качестве точки его завершения. При таком способе реализации необходимо только выполнить проверку — возврат из функции был стартом региона времени или его завершением. Такую проверку легко осуществить по коду возврата из функции.

Ниже представлен возможный вариант использования региона времени в низкоуровневом коде. Функция StartTimeRegion фиксирует начало региона времени. При старте региона времени функция возвращает значение TRUE. Функция FinishTimeRegion фиксирует конец кода региона времени. При возникновении тайм-аута управление передается на точку выхода из функции StartTimeRegion с результатом работы FALSE.

```
struct _TimeRegion *pTimeRegion;
BOOL bResult;
LARGE_INTEGER liTime;
liTime.QuadPart = -10000000*2; // две секунды

// Старт региона времени
bResult = StartTimeRegion(
    SYSTEM_TIME, // Системное время
    liTime,
    &pTimeRegion);

if (bResult)
{
    // Выполняется регион времени
    ...
    // Конец региона времени
    FinishTimeRegion(pTimeRegion);
} else
{    // Конец региона времени
    FinishTimeRegion(pTimeRegion);
    // Действия по тайм-ауту
    ...
}
```

Подобный способ широко используется в операционных системах UNIX для распараллеливания процессов. Функция fork производит создание нового процесса на базе текущего. Оба процесса получают управление в точке выхода из функции.

В случае с регионами времени распараллеливания не происходит, поскольку работа происходит всегда в контексте одной и той же нити. И вход в регион времени, и выход по истечении тайм-аута происходят в точке выхода из функции StartTimeRegion. Различие заключается только в коде возврата функции. Вне зависимости от вида завершения региона времени — по завершении кода региона времени либо по истечении тайм-аута — необходимо выполнить функцию FinishTimeRegion, которая фиксирует окончание региона времени и освобождает память, выделенную для него.

Подобная организация регионов времени весьма удобна для системного программиста. Вход в функцию StartTimeRegion фиксирует адрес, в который нужно передать управление в случае истечения тайм-аута. Этот адрес сохраняется в структуре региона времени. Для прикладного программиста такая организация регионов времени, наоборот,

является весьма неудобной. Отсутствие жесткой связи между стартом региона и его завершением позволяет программисту создавать различные сочетания регионов времени, в том числе и перекресток регионов, которые при выполнении могут давать непредсказуемые результаты. Этого недостатка лишен способ, когда регионы времени интегрируются напрямую в язык программирования.

8 Расширение языка программирования Си для регионов времени

Наиболее предпочтительным способом использования регионов времени для программиста является внедрение регионов времени непосредственно в язык программирования. Ниже представлено предлагаемое расширение языка программирования Си, использующее новые ключевые слова для задания регионов времени.

В рамках расширения внесены следующие изменения в грамматику языка Си: добавлены два новых нетерминала — *try-except-statement* и *try-finally-statement* — в грамматику нетерминала statement. Последняя, в соответствии со стандартом ISO/IEC 9899:TC2 [10] и с учетом внесенных изменений, приведена ниже (добавленные нетерминалы выделены подчеркиванием).

statement:
labeled-statement
compound-statement
expression-statement
selection-statement
iteration-statement
jump-statement
try-except-statement
try-finally-statement
try-timeout-statement

Грамматика нетерминалов *try-except-statement*, *try-finally-statement*, *try-timeout-statement*, а также других связанных с ними конструкций приведена ниже:

try-except-statement:
_try *compound-statement* **_except** (*expression*) *compound-statement*

try-finally-statement:
_try *try-finally-compound-statement* **_finally** *termination-block*

try-timeout-statement: **_try** *timeout-assignment compound-statement* **_timeout** *termination-block*

timeout-assignment:
timeout-specifiers$_{opt}$ (*expression*)

timeout-specifiers:
_system_timeout
_thread_timeout

try-finally-compound-statement:
{ *try-finally-block-item-list*$_{opt}$ }

try-finally-block-item-list:
try-finally-block-item
try-finally-block-item-list try-finally-block-item

try-finally-block-item:
declaration
statement
_leave ;

termination-block:
{ *termination-block-item-list*$_{opt}$}

termination-block-item-list:
termination-block-item
termination-block-item-list termination-block-item

termination-block-item:
declaration
labeled-statement
compound-statement
expression-statement
selection-statement
iteration-statement
jump-statement
try-finally-statement

Помимо изменений в грамматике предлагается внесение изменений в семантику языка Си. Конструкции *try-except* и *try-finally* реализуют «классический» механизм структурной обработки исключений (SHE — Structured Exception Handling), совместимый с аналогичным механизмом, поддерживаемым компилятором Microsoft вместе с операционной системой Windows [11]. Данные расширения носят название ms-specific расширений языка Си/Си++. В качестве расширения механизма SEH предлагается дополнительно ввести блок *try-timeout*.

Тайм-аут блока *try-timeout* определяется выражением, которое при других обстоятельствах могло бы явиться правой частью выражения присвоения (см. грамматику выше). Таким образом, выражение, определяющее тайм-аут блока *try-timeout*, должно быть r-значением (в том числе константным) и иметь тип long long int. Значение данного выражения задает тайм-аут, выраженный в 100-наносекундных интервалах. Тайм-аут может быть задан либо в единицах системного времени с помощью ключевого слова **_system_time** в заголовке блока

try, либо в единицах виртуального времени выполняющей код блока *try* нити с помощью ключевого слова **_thread_time** в заголовке блока *try*. Знак значения выражения, определяющего тайм-аут, задает режим отсчета времени: «+» — абсолютное время; «−» — относительное время. Режим отсчета времени имеет ту же семантику, что подразумевается в других API-функциях, принимающих в качестве одного из своих аргументов значение времени (тайм-аута), например *SetWaitableTimer*. Грамматикой допускается отсутствие спецификаторов режима отсчета в заголовке блока *try*. В этом случае подразумевается спецификатор **_system_time**.

При указании тайм-аута в блоке *try-timeout* и при выполнении кода в блоке *try* за время, меньшее или равное указанному тайм-ауту, управление передается коду, следующему непосредственно за блоком *timeout*. Такая передача управления трактуется как нормальное завершение кода блока *try*. При возникновении тайм-аута или возникновении другой исключительной ситуации управление передается в блок *timeout*. Код в блоке *timeout* способен определить причину передачи управления в блок *timeout*, используя функцию *GetTimeoutCode()*, и предпринять соответствующие действия.

Функция *GetTimeoutCode()* возвращает один из трех кодов завершения: TIMEOUT_EXPIRED, OVERLYING_TIMEOUT_EXPIRED, OVERLYING_EXEPTION_OCCURED. Код TIMEOUT_EXPIRED означает возникновение тайм-аута в текущем блоке. Код OVERLYING_TIMEOUT_EXPIRED означает возникновение тайм-аута в вышележащем блоке. Наконец, код OVERLYING_EXEPTION_OCCURED означает возникновение исключительной ситуации, обработчик которой находится по коду выше текущего блока *try-timeout*. В случае возвращения функцией *GetTimeoutCode()* кода OVERLYING_EXEPTION_OCCURED обработчик может вызвать функцию *GetExceptionCode()* для определения кода исключения.

Если программой заданы вложенные обработчики *finally*, то при выходе из блока *try* как при нормальном завершении, так и при возникновении тайм-аута выполняется код обработчиков *finally*, а также выполняются другие действия, предусмотренные «классическим» механизмом SEH, такие как «раскрутка» (unwinding) и другие.

9 Связь регионов времени с обработчиками исключений

После внесения изменений в язык программирования Си при выполнении программы могут возникнуть следующие комбинации блоков, требующие корректной обработки:

1. Если в процессе работы программы возникает завершение региона времени по тайм-ауту, то необходимо произвести раскрутку обработчиков исключений, для того чтобы остались только те обработчики, которые находятся в стеке SEH выше блока *timeout*.
2. Если в процессе работы программы возникает исключение, а обработчик исключения находится в стеке выше, чем блок *timeout*, отвечающий за обработку региона времени, то в процессе раскрутки стека SEH по возникновении исключения необходимо выполнить код, завершающий регион времени. В противном случае при возникновении тайм-аута стек обработчика региона времени будет уже разрушен.
3. Если в процессе работы программы возникает завершение вышележащего региона времени по тайм-ауту, то необходимо уведомить текущий обработчик, что его регион времени завершается до завершения блока *try*.
4. И при возникновении исключения, и при возникновении тайм-аута необходимо в процессе раскрутки производить выполнение блоков *finally*, расположенных ниже в стеке SEH. Блоки *finally* необходимы для корректного завершения работы с ресурсами программы при возникновении исключения или тайм-аута.

Для корректной работы программы также необходимо, чтобы в течение выполнения блока *finally* механизм регионов времени работал только на отсечку тайм-аутов, а переключения не производились. В противном случае возникнет вероятность некорректной работы программы.

Как видно из вышеприведенного списка, блок *try-timeout* является одной из составных частей стека SEH. При написании программ невозможно отделить регионы времени, оформленные в виде блоков *try-timeout* от других блоков структурной обработки исключений. В противном случае корректная обработка исключений становится невозможной.

10 Стиль написания программ, использующих регионы времени

Если в процессе написания программы возникает необходимость использования блока *try-timeout*, то вся нижележащая работа с ресурсами, включая системные ресурсы и собственные ресурсы программы, должна быть оформлена в виде блоков

try-finally. Использование данных блоков позволяет исключить утечку ресурсов при возникновении тайм-аута.

Ниже приведен пример, иллюстрирующий использование блоков *try-finally* для исключения утечки памяти в процессе работы программы.

```
char *pData = NULL;

    _try _system_time(-10000) { // регион на 1 мс
_try {
    pData = (char*)malloc(512); // выделение памяти
    // работа с памятью
  }
  _finally {
    if (pData)
      free(pData); // освобождение памяти
  }
}
_timeout
{
  printf("Timeout!");
}
```

В блоке *try-finally*, расположенном в составе блока *try-timeout*, происходит работа с памятью. Освобождение памяти помещено в состав блока *finally*. В результате при возникновении тайм-аута в первую очередь будет выполнен код, находящийся в составе блока *finally*, а именно освобождение выделенной памяти. Только после этого управление будет передано блоку *timeout*. Таким образом, потеря памяти при возникновении тайм-аута будет исключена.

Если тайм-аут возникнет, когда программа будет выполняться в блоке *finally*, то переключение на блок *timeout* произойдет только после выхода из блока *finally*.

11 Заключение

Благодаря введению понятия «регион времени» становится возможным придание операционной системе общего назначения черт, изначально присущих только операционным системам реального времени. Регионы времени дают возможность написания более гибкого кода как в составе ядра операционной системы, так и для прикладных программ.

В статье исследованы вопросы аппаратной поддержки регионов времени в современных персональных компьютерах. Рассмотрены принципы и проблемы реализации регионов времени ядром операционной системы.

В статье также предлагается два способа представления регионов времени для программиста: на основе функций и на основе конструкций языка программирования. Второй способ придает большую предсказуемость поведению регионов времени и связывает их с обработчиками исключений, позволяя задавать единый код обработки исключительных ситуаций.

Литература

1. *Соломон Д., Руссинович М.* Внутреннее устройство Microsoft Windows 2000. Мастер-класс / Пер. с англ. — СПб.: Питер; М.: Издательско-торговый дом «Русская редакция», 2001. 752 с.
2. Intel 82371FB (PIIX) AND 82371SB (PIIX3) PCI ISA IDE XCELERATOR. April 1997. — www.intel.com.
3. BIOS and Kernel Developers Guide for AMD Athlon 64 and AMD Opteron Processors. Rev. 3.28. October 2005. — www.amd.com.
4. Intel 64 and IA-32Architectures Software Developers Manual. Vol. 3a: System Programming Guide, Part 1. May 2007. — www.intel.com.
5. Intel 82093AA I/O ADVANCED PROGRAMMABLE INTERRUPT CONTROLLER (IOAPIC). May 1996. — www.intel.com.
6. MultiProcessor Specification Version 1.4. May 1997. — www.intel.com.
7. *Лукач Ю. С., Сибиряков А. Е.* Программно-технические средства персональных ЭВМ семейства IBM PC. — Свердловск: Инженерно-техническое бюро, 1990. 139 с.
8. *Рихтер Дж., Кларк Дж. Д.* Программирование серверных приложений для Microsoft Windows 2000. Мастер-класс / Пер. с англ. — СПб.: Питер; М.: Издательско-торговый дом «Русская редакция», 2001. 592 с.
9. *Дейтел Х. М., Дейтел П. Дж., Чофнес Д. Р.* Операционные системы. Основы и принципы / Пер. с англ. 3-е изд. — М.: ООО «Бином-Пресс», 2007. 1024 с.
10. International Standard ISO/IEC 9899:TC2. May 2005. — www.open-std.org.
11. *Рихтер Дж.* Windows для профессионалов: создание эффективных Win32-приложений с учетом специфики 64-разрядной версии Windows. 4-е изд. — СПб.: Питер; М.: Издательско-торговый дом «Русская редакция», 2001. 752 с.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2008. Т. 2. Вып. 4. С. 85–92

EUROWORDNET: ЗАДАЧИ, СТРУКТУРА И ОТНОШЕНИЯ

О. С. Кожунова

Аннотация: В обзоре приведено краткое описание ресурса EuroWordNet, история его создания, примеры аналогичных ресурсов, а также его задачи, структура и отношения, реализованные в его инструментарии.

Ключевые слова: лексико-семантический ресурс EuroWordNet; «Ворднет-словари» (WordNet); тезаурус; синсет; языковые отношения; межъязыковой индекс ILI

1 Введение

Сегодня для решения задач компьютерной лингвистики привлекается большое количество лексических ресурсов, используемых в сфере информационных технологий.

Одним из наиболее распространенных типов таких ресурсов являются автоматизированные словари, построенные по модели «Ворднет» (WordNet) [1]. Словари типа «Ворднет» объединяют в себе черты справочной системы и инструмента для проведения лингвистических исследований.

В частности, при проведении информационного поиска «Ворднет-словари» удобно использовать для расширения запроса пользователя за счет парадигматически и синтагматически связанных слов, например компонентов синсета (множества синонимов, объединенных в набор) вместе с его гипонимами и согипонимами или связей типа «глагол–актант», которые дают возможность осуществлять контекстный поиск. Данные о синтагматических отношениях слов позволяют применять «Ворднет-словари» для решения задачи снятия неоднозначности смысла слова. «Ворднет» можно использовать для вычисления смысловой близости текстов на основе гиперонимических отношений. «Ворднет-словари» могут служить лексиконом для формальных грамматик. Формат «Ворднет» является удобным формализмом для представления состава и структуры лексики специальных подъязыков (например, медицинских, экономических терминов).

«Ворднет-словари» являются удобным инструментом для проведения исследований в области лексической семантики. Например, гипонимические отношения в «Ворднет-словарях» позволяют определять направление метонимических переносов и прогнозировать появление новых лексико-семантических вариантов [2].

Проект по разработке словаря Princeton WordNet (PWN) английского языка в Принстонском университете (США) стартовал в первой половине 1980-х гг. и продолжается по сей день. Сейчас уже доступна версия WordNet 2.0. Существующая версия охватывает более 120 тыс. слов общеупотребительной лексики современного английского языка [3].

За период с марта 1996 г. по сентябрь 1999 г. при финансировании Европейской комиссии был создан многоязычный вариант WordNet — EuroWordNet [4], что стало новым этапом в эволюции «Ворднет-словарей». В рамках европейского проекта было создано не только несколько тезаурусов для европейских языков (голландского, испанского, итальянского, немецкого, французского, чешского и эстонского), но и впервые была реализована идея об объединении отдельных «Ворднет-представлений» в общую систему. Все компоненты EuroWordNet были построены по единой модели, что, однако, не предполагало прямого перевода английского варианта WordNet 1.5, перед разработчиками стояла задача — отразить все особенности лексических систем национальных языков. Совместимость компонентов EuroWordNet была обеспечена единством принципов и заданным набором общих понятий (Basic Concepts), на основе которых определялась система межъязыковых отсылок (Inter-Lingual-Index, ILI), дающих возможность переходить от лексикализованных значений одного языка к сходным, но не обязательно тождественным значениям в другом языке. Данный индекс позволяет использовать EuroWordNet не только для информационного поиска в рамках одного языка, но и для многоязычного поиска.

В рамках проекта EuroWordNet первоначальная структура словаря претерпела серьезные изменения. Был расширен набор семантических отношений за счет парадигматических

[1]Институт проблем информатики Российской академии наук, okozhunova@ipiran.ru

отношений, связывающих слова разных частей речи (например, XPOS_NEAR_SYNONYMY: dead — death; XPOS_HYPERONYMY: to love — emotion; XPOS_ANTONYMY: to live — dead) и синтагматических отношений между глаголами и актантами-существительными (например, ROLE_INSTRUMENT: to write — pencil). Был сформирован новый подход к построению «Ворднет-словарей»: с опорой на использование лексикографических источников (толковых, переводных и синонимических словарей) и результатов обработки корпусов современных текстов [3].

Успешное завершение проекта EuroWordNet послужило толчком к созданию большого числа «Ворднет-представлений» для языков разных типов (например, венгерского, турецкого, арабского, тамильского, китайского и пр.), а также многоязычных ресурсов типа EuroWordNet (например, проект BalkaNet нацелен на объединение греческого, румынского, болгарского, сербского, турецкого и чешского «Ворднет-словарей»). В 2001 г. была создана Всемирная Ассоциация WordNet (Global WordNet Association), которая ставит целью объединение уже существующих и только развивающихся национальных ресурсов этого типа, усовершенствование системы межъязыковых индексов и разработку общих стандартов, позволяющих использовать модель «Ворднет» для языков разных типов [5].

С 1999 г. на кафедре математической лингвистики СПбГУ исследовательская группа под руководством И. В. Азаровой (О. А. Митрофанова, А. А. Синопальникова и др.) ведет работы по проекту RussNet — созданию русской версии компьютерного словаря типа WordNet. В задачи проекта входит построение лексико-семантического ресурса для отражения организации лексической системы русского языка в целом, для представления ядра его общеупотребительной лексики и фиксации семантических, семантико-грамматических и семантико-деривационных отношений русского языка [3]. Кроме того, в настоящее время в Петербургском университете путей сообщения разрабатывается проект русской версии WordNet под руководством С. А. Яблонского и А. М. Сухонога-ва [6].

В данной статье рассматривается именно лексико-семантический ресурс EuroWordNet, поскольку, в отличие от его главной исходной версии WordNet 1.5 и аналогичных отдельных моделей в других языках, в нем реализована идея объединения отдельных тезаурусов в единую систему, разработан индекс перехода с одного языка на другой с учетом особенностей понятийной системы каждого европейского языка (т. е. осуществляется возможность перехода от лексикализованных значений одного языка к аналогичным, но не обязательно тождественным понятиям другого языка). Такие особенности EuroWordNet наделяют этот словарь качественно иной функциональностью, что обусловлено спецификой его структуры и системы отношений.

2 Общая характеристика проекта EuroWordNet

Цель проекта EuroWordNet, как было сказано во введении, состоит в построении многоязычной лексической базы данных с «Ворднетами» для нескольких европейских языков, которые структурированы согласно принципам Princeton WordNet (далее — WordNet) [4]. WordNet содержит информацию о существительных, глаголах, прилагательных и наречиях английского языка. Одним из его базовых понятий является синсет (synset). Синсет представляет собой набор слов, принадлежащих к одному типу частей речи, которые взаимозаменямы в определенном контексте.

Например, слова из множества {тачка; авто; автомобиль; машина; автомашина} формируют синсет, поскольку их можно использовать для обращения к одному и тому же понятию. Далее часто приводится толкование синсета: «Четырехколесный; обычно работающий от двигателя внутреннего сгорания». Синсеты могут быть связаны друг с другом семантическими отношениями, такими как гипонимия (между конкретными и более общими понятиями), меронимия (между частями и целыми), причинно-следственные отношения и т. д., как показано на рис. 1. В этом примере, который заимствован из WordNet 1.5 [1, 2], синсет {тачка; авто; автомобиль; машина; автомашина} связан с:

(1) более общими понятиями или синсетом гиперонимов: {автомобили; механическое транспортное средство};

(2) более конкретными понятиями или синсетом гипонимов: например, {патруль; полицейская автомашина; патрульная спецмашина; полицейский автомобиль; патрульный полицейский автомобиль общего назначения} и {кеб; такси; машина для извоза; такси-кеб};

(3) частями, из которых он состоит: например, {бампер}, {дверца автомобиля}, {автомобильное зеркало} и {окно автомобиля}.

Каждый из этих синсетов далее связан с другими синсетами. Это хорошо видно из примера, где синсет {автомобили; транспортное механическое

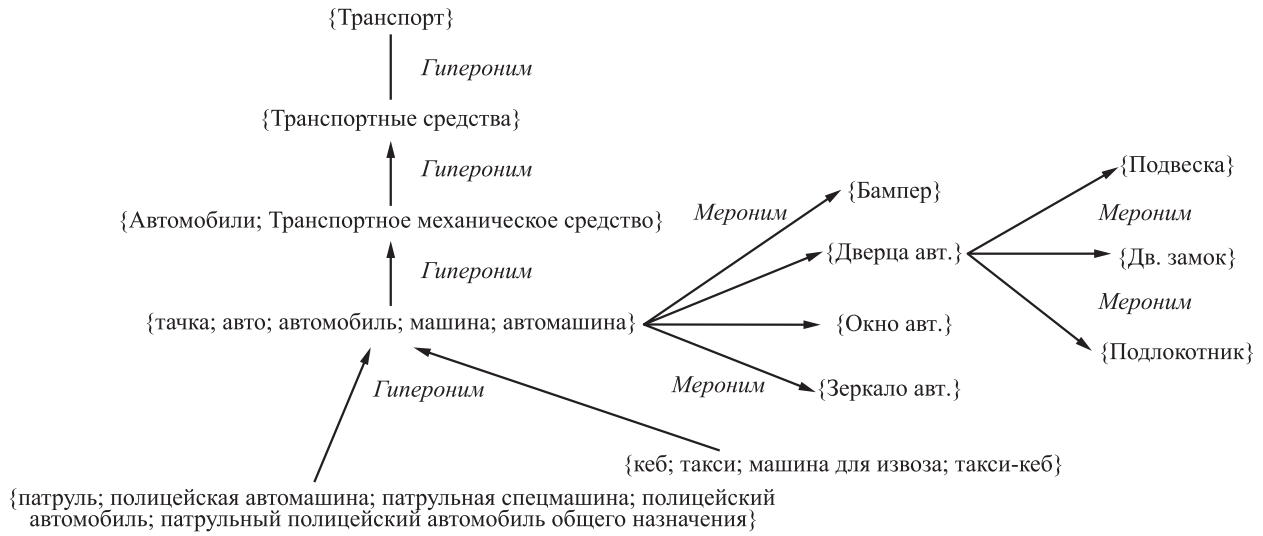


Рис. 1 Синсеты, связанные с понятие «машина» в его первом смысле в WordNet 1.5

средство} связан с синсетом {транспортные средства}, а синсет {дверца авт.} связан с синсетами, содержащими упоминания других частей автомобиля: {подвеска}, {подлокотник}, {дв. замок} [1].

Посредством этих и других семантических (концептуальных) отношений могут быть установлены связи между всевозможными значениями, составляющими при этом огромную сеть или «Ворднет».

Такой «Ворднет» может быть использован для формирования суждений о значениях слов (какие именно значения могут быть интерпретированы как «транспортные средства») и для осуществления информационного поиска. В частности, для поиска альтернативных выражений или формулировок или просто с целью расширения словарных множеств до наборов семантически связанных или близких слов. Кроме того, семантические сети дают информацию о лексикализованных шаблонах, о концептуальной плотности областей словарей и о распределении семантических различий или отношений по разным областям словарей различных языков [4].

Ресурсы европейских «Ворднетов» будут храниться в центральной лексической системе баз данных, причем каждое лексическое значение будет связано с наиболее близким синсетом принстонского WordNet 1.5, формируя таким образом многоязычную базу данных [1, 4].

В такой базе данных станет возможным переход от одного значения из какого-либо «Ворднета» к значению в другом «Ворднете», который связан с аналогичным понятием из WordNet 1.5. Эта многоязычная база данных может стать полезной для кроссязыкового поиска информации, для передачи информации из одного ресурса в другой или для простого сравнения различных «Ворднетов» [7]. Сравнение может дать информацию о непротиворечивости отношений в «Ворднетах», различия в которых могут указывать на противоречия или на специфичные для некоторого языка свойства ресурсов, а также на свойства самого языка [4].

Таким образом, базу данных можно также рассматривать как мощное средство для изучения лексических семантических ресурсов и их языковой специфики [8].

В EuroWordNet ее разработчики [1, 4, 7–10] первоначально имели дело с четырехязыковой моделью. Первыми языками, для которых были разработаны вышеописанные ресурсы, стали голландский, итальянский, испанский и английский. Размер каждого из этих «Ворднетов», за исключением «Ворднета» английского языка, составляет около 30 000 сравнимых синсетов, что приблизительно соответствует 50 000 лексических значений. Для сравнения: размер WordNet 1.5 составляет 91 591 синсетов и 168 217 значений слов. В качестве развития проекта его база данных была расширена соответствующими ресурсами немецкого, французского, эстонского и чешского языков. Размер этих «Ворднетов» колеблется в диапазоне от 7 500 до 15 000 синсетов.

Состав «Ворднетов» ограничен существительными и глаголами, хотя и прилагательные, и наречия включены при существовании связи с существительными и глаголами. В словарь системы будут включены все общие и основные слова языков. Таким образом, он будет включать все значения и понятия, которые необходимы для связывания более конкретных значений и всех слов, наиболее часто встречающихся в общих текстовых

корпусах. Для одной из областей будет добавлен вспомогательный словарь для иллюстрации возможности объединения различной терминологии в таком универсальном словаре.

3 Комплексная модель базы данных EuroWordNet

Разработка базовых семантических ресурсов представляет собой нетривиальную задачу. Смысл и его толкование изучаются множеством дисциплин и парадигм с различными точками зрения и подходами. Широко распространено мнение, что роль общей семантики, или изучения процессов понимания языка, все еще чрезмерно сложна для ее описания, адекватного современным технологиям. Поэтому цель проекта EuroWordNet состоит не в том, чтобы разрабатывать полные семантические словари, которые используют сложные языковые представления и механизмы логического вывода, а в том, чтобы свести определенную языковую информацию и механизмы к тем базовым семантическим отношениям между словами, которые хорошо изучены [1, 4, 10].

Семантические отношения, встроенные в WordNet 1.5 признаны во многих научных областях: формальной семантике, искусственном интеллекте, когнитивной лингвистике, лексикологии, информатике и математике [1]. Кроме того, отношения не основываются на каком-либо специфичном формализме представления знаний. Предполагается, что они будут основой любой системы знаний будущего [4]. Даже при том, что реляционный подход к смыслу в WordNet и EuroWordNet нельзя рассматривать как полное описание смысла слова, тем не менее в этих ресурсах предусмотрена возможность расширения такой базовой информации более исчерпывающими описаниями с частичным использованием корпусно-текстовых технологий [7–9]. Более того, в работе Sanfilippo [11] продемонстрировано, что доступность «Ворднет-структур» очень полезна для автоматического извлечения такой информации из корпусов текстов.

Таким образом, проект EuroWordNet-database — это первый проект из всевозможных аналогичных разработок, базирующихся на структуре PWN, в частности на версии WordNet 1.5. Из этой версии для проекта EuroWordNet были заимствованы понятие синсета и базовые семантические отношения [4].

Однако в проект базы данных были внесены определенные изменения, которые в основном мотивированы следующими факторами [4]:

(1) стремлением создать многоязычную базу данных;

(2) потребностью в поддержании ориентированных на конкретный язык отношений в «Ворднет»;

(3) стремлением достигнуть максимальной совместимости с другими ресурсами;

(4) необходимостью в построении «Ворднетов» с использованием существующих ресурсов в относительно независимом режиме.

Поэтому важнейшим отличием EuroWordNet от WordNet является его многоязычность, в связи с которой, тем не менее, также возникают некоторые принципиальные вопросы в отношении статуса моноязычной информации, отраженной в «Ворднетах» [7]. Многоязычность достигается, главным образом, добавлением отношения эквивалентности для каждого синсета некоторого языка с наиболее близким ему синсетом из WordNet 1.5. Предполагается, что синсеты, связанные с одним и тем же синсетом из WordNet 1.5, эквивалентны или близки по смыслу, а следовательно, сравнимы. Однако возникает проблема различий «Ворднетов». Если «эквивалентные» слова по-разному связаны в различных ресурсах, то необходимо принять решение о законности этих различий [4]. Например, в голландском «Ворднете» hond (“dog” в WordNet 1.5) классифицируется и как huisdier (“pet” в WordNet 1.5), и как zoogdier (“mammal” в WordNet 1.5). Тем не менее в итальянском языке нет никакого эквивалента для «домашнего животного» (“pet” в WordNet 1.5), а слово “cane”, которое связано с тем же самым синсетом “dog”, в итальянском «Ворднете» классифицируется только как “mammal” [4].

Разработчики EuroWordNet придерживаются мнения, что необходимо сделать все возможное для отражения подобных различий в лексических семантических отношениях. «Ворднеты» скорее рассматриваются в качестве лингвистических онтологий, чем только как онтологии для производства выводов [1, 8]. В онтологии, основанной на выводах, может быть так, что для достижения лучшего управления, или результативности, или компактной и последовательной структуры необходим определенный подход к структурированию. В связи с этим может возникнуть необходимость во введении искусственных уровней для понятий, которые не выражены словами в языке (например, природные объекты, внешние части тела), или в исключении уровней (например, сторожевой пес), которые представлены словами языка, но нерелевантны целям онтологии. С другой стороны, лингвистическая онтология точно отражает лексикализацию

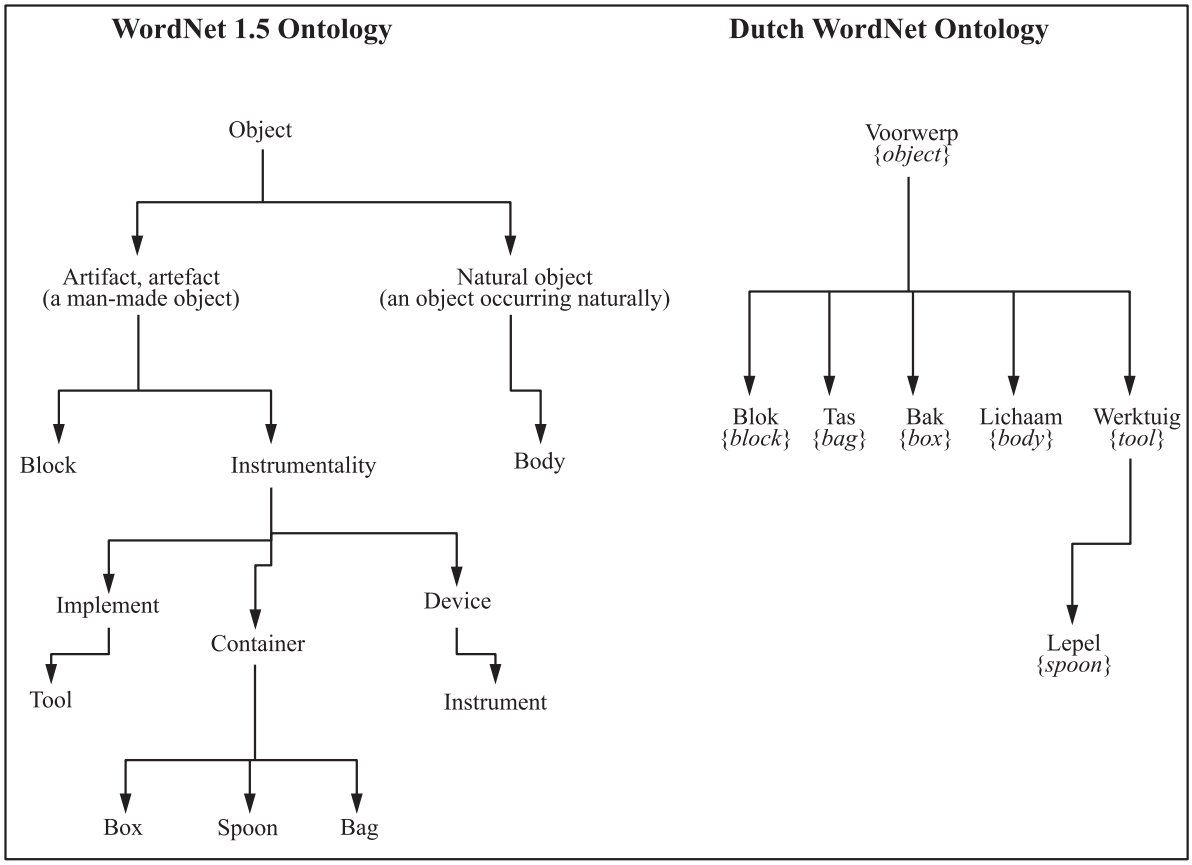


Рис. 2 Лексикализованные и нелексикализованные уровни в «Ворднетах»

(т. е. представление понятий словами) и отношения между словами в языке. Это «Ворднет» в истинном смысле этого слова и, следовательно, охватывает ценную информацию о представлениях, которые лексикализованы в языке, а именно информацию о том, каков объем доступных слов и выражений некоторого языка [4].

Данное отличие проиллюстрировано на рис. 2, где гипонимическая структура WordNet 1.5 содержит сочетание лексикализованных и нелексикализованных категорий, в то время как голландский «Ворднет» содержит только лексикализованные категории языка [1, 4, 11].

На примере фрагмента WordNet 1.5, приведенного на рис. 2, видно, что синсет для понятия «объект» (object) сначала подразделяется на два подкласса «артефакт» (artefact) и «природный объект» (natural object), из которых последний является нелексикализованным выражением в английском языке (которое должно быть в словаре языка в качестве заглавного слова), а скорее представляет собой построенное в соответствии с правилами языка выражение [4]. У класса «артефакт» (artefact) есть важный подкласс «инструментарий» (instrumentality), который используется для группировки таких взаимосвязанных синсетов, как «орудие» (implement), «устройство» (device), «инструмент» (tool) и «средство» (instrument). Создается впечатление, что такая группировка может быть полезной для организации иерархии и прогнозирования функциональности подклассов. Тем не менее она не обеспечивает корректными прогнозами о взаимозаменяемости существительных, т. е. нельзя обратиться к существительным «контейнеры» (containers), «коробки» (boxes), «ложки» (spoons) и «мешки» (bags), используя существительное «инструментарий» (instrumentality) английского языка [1, 4].

На примере фрагмента голландской иерархии, приведенного на рис. 2, видно, что такие искусственные уровни, как «природный объект» (natural object) и «инструментарий» (instrumentality) использованы не были. Кроме того, точных эквивалентов для английских “artifact” и “container” в голландском языке нет. В результате получается гораздо более плоская иерархия, в которой такие специфичные свойства, как «натуральный» (natural), «искусственный» (artificial) и «функциональность»

(functionality) не могут быть выведены из отношений гипонимии [1, 4].

С другой стороны, такая сеть предоставляет корректные прогнозы выразительных возможностей словаря голландского языка, поскольку включает в себя только допустимые для языка слова (и составные выражения). Можно было бы построить новые классы и выражения голландского языка, чтобы охватить различные обобщения в языке, но априорного критерия для определения полезных или ненужных классов не существует [4]. Можно было бы, наконец, добавить в такую сеть какое-либо потенциальное семантическое свойство в виде нового класса с целью создания детальных структур наследования или заимствовать всевозможные классификации из всех других «Ворднетов». Однако подобные модификации уничтожили бы данный «Ворднет» как сеть допустимых выражений языка. Кроме того, эти изменения не гарантируют наличия хорошей концептуальной онтологии для наследования свойств [4].

Далее можно расширять базу данных посредством отдельных, нейтральных в плане используемых языков или специализированных онтологий, которые содержат адекватные механизмы вывода и хорошо спроектированы для этой цели. Если такая онтология привязана к базе данных EuroWordNet, все «Ворднеты» могут получать доступ к этим классификациям с целью поиска правильных выводов для их синсетов. В таком случае «Ворднеты» обеспечивают точное отображение ориентированного на конкретный язык словаря этой онтологии [4, 8].

Для поддержания целостности ориентированных на конкретный язык структур и отдельной модификации независимых ресурсов разработчики EuroWordNet проводят различие между ориентированными на конкретный язык модулями и отдельным независимым от языка модулем. Каждый модуль языка представляет собой автономную и уникальную, ориентированную на конкретный язык систему внутриязыковых отношений между синсетами. Отношения эквивалентности между синсетами различных языков и синсетами WordNet 1.5 предельно четко выражены при помощи так называемого межъязыкового индекса (ILI). Каждый синсет в моноязычных «Ворднетах» будет связан, по крайней мере, одним отношением эквивалентности с записью данного ILI. Поэтому ориентированные на конкретный язык синсеты, которые связаны с той же самой записью ILI, должны быть эквивалентными для всех языков. Это проиллюстрировано на рис. 3 примером для ориентированных на конкретный язык синсетов, связанных с записью ILI “drive” (ехать, вести) [4, 7–9].

На рис. 3 приведено схематическое представление различных модулей и отношений между ними. В середине рисунка показаны модули, общие для всех языков: ILI (межъязыковой индекс), Онтология предметной области и Онтология понятий верхнего уровня. ILI состоит из списка так называемых записей ILI (ILIRs), которые связаны со значениями слов из модулей конкретных языков, (возможно) с одним или более понятий верхнего уровня и (возможно) с предметными областями. Таким образом, модули конкретных языков состоят из схемы лексических единиц этих языков, связанных индексами с наборами синсетов, между которыми построены внутриязыковые отношения [4, 7].

Индекс ILI представляет собой неструктурированный список значений, заимствованный в основном из WordNet 1.5, где каждая запись ILI состоит из некоторого синсета, толкования на английском языке, определяющего конкретное значение, и ссылку на его источник. Единственная цель ILI заключается в связывании синсетов из ориентированных на конкретный язык «Ворднетов». Поэтому между записями ILI никаких отношений как таковых не предусмотрено [4].

Некоторые из процедур структурирования ILI, независимого от выбора языка, обеспечиваются двумя отдельными онтологиями, которые могут быть связаны с записями ILI [4, 9]:

- онтологией понятий верхнего уровня, которая представляет собой иерархию независимых от выбора языка понятий, отражая важные семантические различия (например, различия между понятиями «объект» и «вещество», «динамический» и «статический»);
- иерархией наименований предметных областей, которые являются структурами знаний, группирующими значения в терминах различных сфер или ситуаций (например, «движение», «дорожное движение», «воздушное движение», «спорт», «больница», «ресторан»).

Как понятия верхнего уровня, так и наименования предметных областей, могут быть переданы при помощи отношений эквивалентности между записями ILI и языковыми значениями конкретного языка, что проиллюстрировано на рис. 3.

Например, понятия верхнего уровня «позиция, местоположение» (location) и «динамический» (dynamic) непосредственно связаны с записью ILI «ехать, везти» (drive). Следовательно, они также имеют косвенное отношение ко всем понятиям конкретных языков, связанных с этой записью ILI. Далее посредством внутриязыковых отношений понятия верхнего уровня могут наследоваться

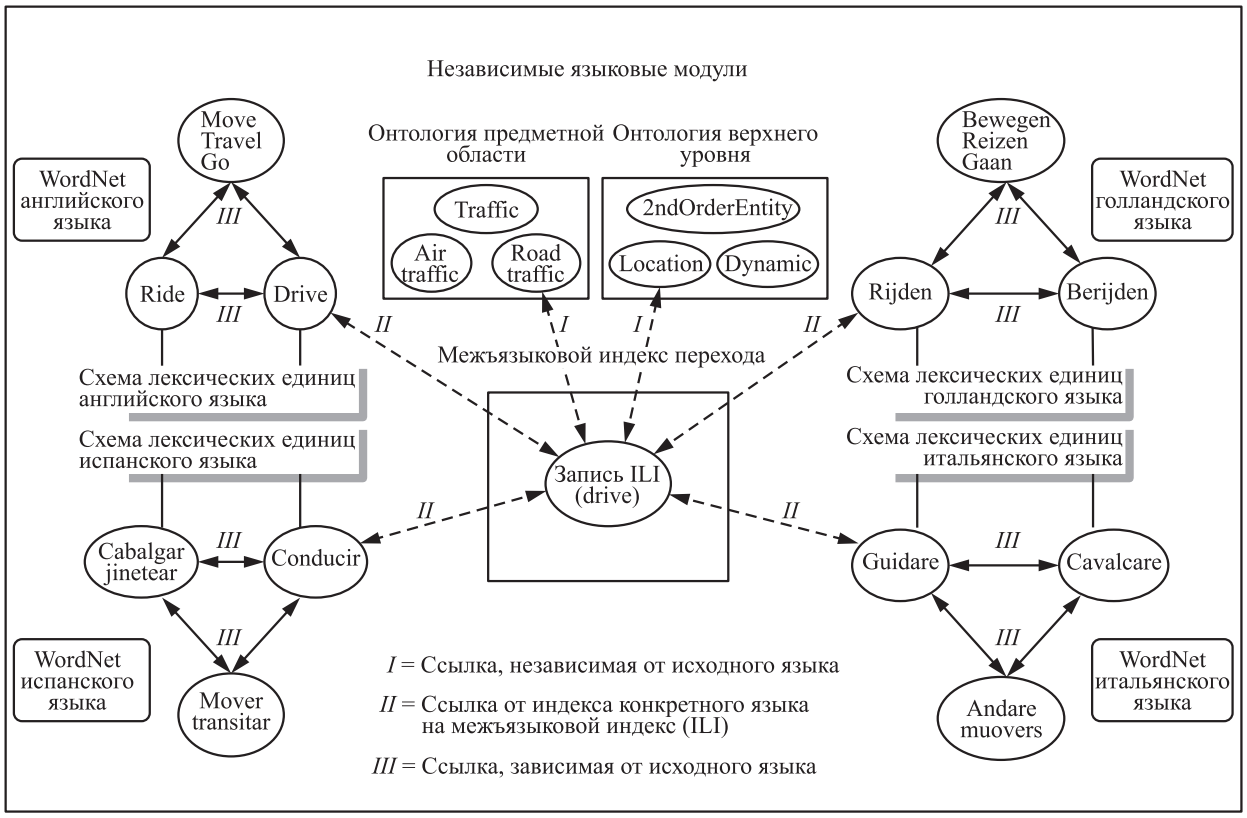


Рис. 3 Архитектура базы данных EuroWordNet

всеми другими соответствующими понятиями конкретных языков [4].

Основная цель онтологии верхнего уровня состоит в формировании общей структуры для важнейших понятий всевозможных «Ворднетов». Данная онтология содержит 63 основных семантических различия, которые классифицируют множество 1024 записей ILI, представляющих важнейшие понятия из различных «Ворднетов» [9].

Наименования предметных областей могут быть применены непосредственно при информационном поиске (а также в инструментариях для изучения языка и при публикации словарей) к групповым понятиям, основанным скорее на предписаниях, чем на классификациях. Предметные области также могут быть использованы для разделения словарей общего назначения и словарей конкретных областей. Это играет важную роль при решении проблемы двусмысленности при обработке естественно-языковых текстов [7].

В проекте EuroWordNet онтология предметной области будет построена только для небольшого фрагмента словаря с целью иллюстрации. Тем не менее пользователи базы данных проекта смогут свободно добавлять наименования предметных областей к межъязыковому индексу ILI или уточнять понятия онтологии верхнего уровня без доступа или рассмотрения внутриязыковых отношений каждого «Ворднета». Таким же образом возможно расширение этой базы данных при помощи других онтологий при условии, что они определены в соответствии с форматом EuroWordNet и содержат надлежащую ссылку на межъязыковой индекс ILI [9].

4 Заключение

Описанный в статье модульный многоязычный проект EuroWordNet имеет следующие достоинства [4]:

- возможность использования базы данных для многоязычного поиска информации посредством расширения набора слов одного языка соответствующими словами другого языка через межъязыковой индекс ILI;
- различные «Ворднеты» могут быть подвергнуты сравнению и кросслингвистической проверке, что сделает их более совместимыми;
- особенности конкретных языков можно реализовать в отдельных «Ворднетах»;

– возможность разработки «Ворднетов» на различных сайтах Интернета в относительно независимом режиме;
– независимая от выбора исходного языка информация, такая как толкования, знания предметной области и аналитические понятия верхнего уровня, может быть сохранена только один раз и может стать доступной для всех модулей, предназначенных для определенных языков, посредством поддержания межъязыковых отношений;
– база данных проекта может быть адаптирована к потребностям пользователей при помощи модификации понятий верхнего уровня, наименований предметных областей или их сущностей (например, добавлением семантических признаков) без необходимости доступа к конкретным языковым «Ворднетам».

Помимо многоязычного дизайна базы данных проекта были осуществлены некоторые изменения во внутриязыковых отношениях по отношению к исходному проекту WordNet 1.5, а именно [4, 7]:

(1) использование наименований отношений, что делает семантические следования более явными и точными;
(2) введение общих отношений между частями речи, так что может быть найдено соответствие между другими поверхностными реализациями подобных понятий в самом языке и его пересечениях с другими языками;
(3) добавление некоторых новых отношений с целью поиска различений определенных поверхностных иерархий.

Литература

1. *Fellbaum C.* WordNet: An electronic lexical database. — Cambridge, 1998.
2. *Miller G., Beckwith R., Fellbaum C., Gross D., Miller K.* Five papers on WordNet // CSL Report 43. — Princeton University, Cognitive Science Laboratory, 1990.
3. *Азарова И. В., Митрофанова О. А., Синопальникова А. А., Ушакова А. А., Яворская М. В.* Разработка компьютерного тезауруса русского языка типа WordNet // Доклады научной конференции «Корпусная лингвистика и лингвистические базы данных» / Под ред. А. С. Герда. — СПб., 2002. С. 6–18.
4. *Vossen P.* Introduction to EuroWordNet // Computers and the Humanities, Special Issue on EuroWordNet, 1998. Vol. 32. No. 2–3. P. 73–89.
5. *Азарова И. В., Синопальникова А. А., Яворская М. В.* Принципы построения wordnet-тезауруса RussNet // Материалы конференции Диалог-2004. — М.: Наука, 2004.
6. *Сухоногов А. М., Яблонский С. А.* Словари типа WordNet в технологиях Semantic Web // Девятая Национальная конференция по искусственному интеллекту с международным участием КИИ-2004. Тр. конференции в 3-х т. — М.: Физматлит, 2004. Т. 2. С. 557–564.
7. *Gonzalo J., Verdejo F., Peters C., Calzolari N.* Applying EuroWordNet to cross-language text retrieval // Computers and the Humanities, Special Issue on EuroWordNet, 1998. Vol. 32. No. 2–3. P. 185–207.
8. *Alonge A., Calzolari N., Vossen P., Bloksma L., Castellon I., Marti T., Peters W.* The linguistic design of the EuroWordNet Database // Computers and the Humanities, Special Issue on EuroWordNet, 1998. Vol. 32. No. 2–3. P. 91–115.
9. *Rodriguez H., Climent S., Vossen P., Bloksma L., Peters W., Roventini A., Bertagna F., Alonge A.* The top-down strategy for building EuroWordNet: Vocabulary coverage, base concepts and top ontology // Computers and the Humanities, Special Issue on EuroWordNet, 1998. Vol. 32. No. 2–3. P. 117–152.
10. *Vossen P., Bloksma L., Peters C., Alonge A., Roventini A., Marinai E., Castellon I., Marti T., Rigau G.* Compatibility in interpretation of relations in EuroWordNet // Computers and the Humanities, Special Issue on EuroWordNet, 1998. Vol. 32. No. 2–3. P. 153–184.
11. *Sanfilippo A.* Using semantic similarity to acquire co-occurrence restrictions from corpora // ACL/EACL'97 Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications Proceedings / Eds. P. Vossen, N. Calzolari, G. Adriaens, A. Sanfilippo, Y. Wilks. Madrid, 1997.

РЕЦЕНЗИИ

МОНОГРАФИЯ А. С. ШАЛАМОВА «ИНТЕГРИРОВАННАЯ ЛОГИСТИЧЕСКАЯ ПОДДЕРЖКА НАУКОЕМКОЙ ПРОДУКЦИИ» (М.: Университетская книга, 2008. 464 с.)

Заслуженный деятель науки РФ, д.т.н., профессор И. Н. Синицын

В июне текущего 2008 г. в издательстве «Университетская книга» вышла монография «Интегрированная логистическая поддержка наукоемкой продукции».

Ее автор Шаламов А. С., профессор, доктор технических наук, заместитель руководителя Департамента послепродажного обслуживания авиационной техники ОАО «Российская самолетостроительная корпорация «МиГ», возглавляет направление работ по интегрированной логистической поддержке (ИЛП).

Интегрированная логистическая поддержка — организационно-технологический и программно-технический комплекс *управления послепродажным обслуживанием* наукоемкой продукции (НП), направленный на создание и обеспечение функционирования интегрированной информационной среды, объединяющей всех участников жизненного цикла (ЖЦ) НП в единое виртуальное предприятие, с целью *минимизации финансовых затрат при заданном уровне технической (эксплуатационной) готовности* парка изделий, что и является главным принципом ИЛП.

Помимо информационного обеспечения весьма важными являются задачи *интеллектуальной поддержки* управления послепродажным обслуживанием НП, на которую и возлагается реализация указанного принципа. Недостаточный уровень существующих методов математического моделирования не позволяет пока осуществить это в полной мере. Основной целью рецензируемой монографии как раз и является разработка теоретических и практических подходов к решению подобных задач.

В первых двух частях книги после краткого описания системы послепродажного обслуживания (СППО) как объекта моделирования и управления рассматриваются различные вопросы — от процессов управления конфигурацией изделий, СППО на этапах жизненного цикла и моделирования стоимости жизненного цикла НП, до информационного моделирования основных логистических процессов с использованием современных международных стандартов, включая демонстрацию программных решений при создании интегрированной логистической базы данных.

В третьей, основной части работы, излагаются современные инновационные технологии в области математического моделирования и оптимизации организационно-технических и организационно-экономических систем (ОТС-ОЭС), имеющие российский приоритет. Приведенные здесь научные методы разработаны впервые. С опорой на известные результаты теории непрерывно-дискретных марковских процессов получено интегро-дифференциально-разностное уравнение типа Колмогорова–Феллера для плотности вероятности фазового вектора применительно к СППО с конечным множеством дискретных состояний объектов обслуживания, изменяющихся в течение непрерывно-дискретного времени. Получено также соответствующее уравнение Пугачёва для характеристической функции фазового вектора СППО, используемое для вывода дифференциальных уравнений, определяющих математическое ожидание и ковариационную матрицу вектора фазовых координат, а также вероятность эффективной работы системы. В основе уравнений лежит предложенный автором способ формализации организационно-технологической структуры СППО как сети систем массового обслуживания с нелинейными и нестационарными функциями производительности (пропускной способности). Достаточно большое место отводится вопросам адаптации существующих методов оптимизации динамических систем применительно к нелинейной СППО в условиях нестационарного стохастического характера ее непрерывно-дискретных процессов и влияния внешней среды.

Для большей конкретизации можно указать на теоретические результаты в области моделирования вероятностной динамики СППО с различными стратегиями расходования, восстановления и пополнения ресурсов в комплексе с другими процессами виртуального предприятия, а также методы оптимизации как параметров СППО на этапе проектирования НП, так и процессов управления при ее эксплуатации. Интерес представляют также результаты в области систем с ограниченной областью функционирования (изменения фазового вектора). Выход одной или нескольких фазовых координат на границы допустимой области здесь может означать прекращение функционирования системы. В работе предложена модель такой системы, в основе которой (модели) лежит элемент поглощения реализаций фазового вектора на границе области.

В практическом плане на основе разработанных теоретических методов решен ряд задач интеллектуальной поддержки проектирования НП по критерию конкурентоспособности (рыночного потенциала), оптимального планирования потребностей заказчика в ресурсах, необходимых для эксплуатации НП на предстоящем временно́м периоде, а также по определению оптимальных параметров политики поставок для по-

полнения складов как потребителя, так и поставщика, что нашло отражение в двух последних частях монографии.

Предложенные в работе теоретические методы и проведенные прикладные исследования являются основой при создании современных программных комплексов для автоматизированных систем управления широким классом ОТС-ОЭС, включающих в себя практически весь набор подсистем, предназначенных для обслуживания и сопровождения как изделий наукоемкой продукции военного и гражданского назначения, так и объектов материальной инфраструктуры, таких как, например, технические комплексы и объекты машиностроения, энергетики и др.

Кроме того, полученные результаты применимы для создания эффективных систем моделирования процессов, прогнозирования результатов деятельности и оптимального управления в банковской, страховой, лизинговой и других сферах экономики.

Также одной из важнейших проблем по реализации современных методов управления СППО является необходимость организации системы фундаментальной подготовки кадров в этой весьма специфической области знаний, находящейся на стыке многих наук, таких, по крайней мере, как техническая эксплуатация НП, прикладная математика и информатика. Монография Шаламова А. С., являясь своего рода учебным пособием, содержит основы для решения и этой задачи, включая подготовку аспирантов и докторантов с целью обеспечения притока вузовских кадров (педагогов и ученых).

МОНОГРАФИЯ И. Н. СИНИЦЫНА «ФИЛЬТРЫ КАЛМАНА И ПУГАЧЕВА» (Изд. 2-е перераб. и доп., М.: Университетская книга, ЛОГОС, 2007. 776 с.)

Член-корреспондент РАН А. П. Реутов

В книге дано систематическое изложение теории фильтров Калмана и Пугачёва для обработки информации в сложных стохастических системах, а также приведены новые результаты фундаментальных работ, выполненных в Институте проблем информатики Российской академии наук в рамках научного направления «Стохастические системы и стохастические информационные технологии».

Во втором издании книга подверглась существенной переработке с целью ориентации на читателя, знакомого только с элементарной теорией вероятностей и математической статистики. В отдельные главы выделены собственно теория фильтров Калмана и Пугачёва, а также некоторые прикладные задачи оценивания, распознавания и идентификации сигналов и параметров на основе фильтров Калмана и Пугачёва.

Глава 1 содержит необходимые сведения по теории случайных величин и функций. Изложены основы стохастического анализа.

В гл. 2 приведены необходимые сведения по моделям непрерывных и дискретных стохастических систем (СтС). Рассмотрена теория одно- и многомерных распределений процессов в СтС. Описаны элементы теории оценивания в непрерывных и дискретных СтС.

В гл. 3 рассмотрен фильтр Калмана для непрерывных и дискретных линейных СтС. Изложены элементы линейного стохастического анализа непрерывных СтС. Выведены основные уравнения оптимальной фильтрации в гауссовских непрерывных СтС. Особое внимание уделено уравнениям, линейным относительно вектора состояния. Рассмотрена теория непрерывных фильтров и экстраполяторов. Даны обобщения калмановской теории фильтрации на случай автокоррелированной помехи в наблюдениях. Специальный раздел посвящен вопросам устойчивости фильтра Калмана–Бьюси. Изложена теория дискретного фильтра Калмана.

Глава 4 посвящена теории приближенных (субоптимальных) методов оценивания состояния и параметров в нелинейных СтС, основанная на теории нелинейного оценивания. Приведены элементы нелинейного стохастического анализа непрерывных СтС, основанные на методах нормальной аппроксимации (МНА), эквивалентной линеаризации, а также методах параметризации распределений. Дана краткая характеристика субоптимальных методов оценивания для дифференциальных СтС. Подробно рассмотрены МНА апостериорного распределения и метод статистической линеаризации (МСЛ). Описан модифицированный МНА, основанный на использовании ненормированных распределений. Особое внимание уделено квазилинейным субоптимальным фильтрам, основанным на МСЛ. Приведены методы моментов, семиинвариантов, ортогональных разложений и квазимоментов для приближенного решения фильтрационных уравнений, а также модифицированные версии методов, основанные на использовании ненормированных распределений. Подробно рассмотрены квазилинейные субоптимальные методы оценивания, основанные на методах параметризации распределений. Специальный раздел отведен методам субоптимального оценивания, основанным на упрощении уравнений оптимальной фильтрации. Большое внимание уделено непрерывному обобщенному фильтру Калмана (ОФК), а также дискретному ОФК. Рассмотрены дискретные субоптимальные фильтры, основанные как на приближенном решении фильтрационных уравнений, так и на их упрощениях.

Глава 5 содержит систематическое изложение теории фильтра В. С. Пугачёва. Изложен принцип условно оптимальной фильтрации и постановки основных задач. Дано решение задач условно оптимальной фильтрации, экстраполяции и интерполяции. Рассмотрены фильтрация при автокоррелированной помехе в наблюдениях, линейная фильтрация Пугачёва.

В гл. 6 рассмотрены некоторые прикладные задачи оценивания, распознавания и идентификации на основе фильтров Калмана и Пугачёва, фильтры Пугачёва для линейных СтС с параметрическими шумами, фильтры Калмана и Пугачёва по бейесовым и сложно статистическим критериям. Излагаются элементы эллипсоидального анализа распределений в СтС, а также теория субоптимальных и условно оптимальных фильтров для задач фильтрации, распознавания и идентификации сигналов и параметров в нелинейных СтС. Рассмотрено применение фильтров Калмана и Пугачёва в задачах совместной фильтрации, распознавания и идентификации.

В приложениях 1–5 содержатся сведения о полиномах Эрмита, χ^2-распределении и полиномах, ортогональных к χ^2-распределению, функции Лапласа и ее производных, а также формулы для статистической и эллипсоидальной линеаризации. В приложении 6 приведены сведения по известному программному обеспечению фильтров Калмана и Пугачёва, а также примеры его использования.

Биографические замечания и список литературных источников даны в конце книги. Автор в конце биографических замечаний счел необходимым привести портреты и краткие биографические сведения о Р. Э. Калмане (р. 1930) и В. С. Пугачёве (1911–1998).

Книга предназначена для научных работников и инженеров в области прикладной математики и информатики, системного анализа, теории управления, а также в других областях науки и техники, связанных с обработкой информации в системах, поведение которых описывается стохастическими дифференциальными, интегральными, интегродифференциальными, разностными и другими уравнениями (стохастические системы). Книга может представлять интерес для математиков, специализирующихся в области стохастических уравнений и их приложений. Она может быть полезна студентам высших учебных заведений, обучающихся по специальности «Прикладная математика и информатика». Единая методика, тщательный подбор примеров и задач (их свыше 500) позволяют использовать книгу широкому кругу студентов, аспирантов и преподавателей.

МОНОГРАФИЯ В. Е. БЕНИНГА, В. Ю. КОРОЛЕВА, И. А. СОКОЛОВА, С. Я. ШОРГИНА «РАНДОМИЗИРОВАННЫЕ МОДЕЛИ И МЕТОДЫ ТЕОРИИ НАДЕЖНОСТИ ИНФОРМАЦИОННЫХ И ТЕХНИЧЕСКИХ СИСТЕМ» (М.: ТОРУС ПРЕСС, 2007. 256 с.)

Д.ф.-м.н., профессор А. В. Печинкин

В 2007 г. в издательстве «ТОРУС ПРЕСС» вышла монография «Рандомизированные модели и методы теории надежности информационных и технических систем». Ее авторы — В. Е. Бенинг, профессор, доктор физико-математических наук, профессор факультета вычислительной математики и кибернетики МГУ им. М. В. Ломоносова, старший научный сотрудник Института проблем информатики РАН; В. Ю. Королёв, профессор, доктор физико-математических наук, профессор факультета вычислительной математики и кибернетики МГУ им. М. В. Ломоносова, ведущий научный сотрудник Института проблем информатики РАН; И. А. Соколов, академик, директор Института проблем информатики РАН; С. Я. Шоргин, профессор, доктор физико-математических наук, заместитель директора Института проблем информатики РАН.

Для исследования надежности информационных и технических систем (ИиТС), подверженных влиянию случайных и нестационарно изменяющихся факторов, в книге предложены альтернативные классическим рандомизированные математические модели и методы. Большой интерес представляет непараметрический подход к оцениванию коэффициента готовности, для получения интервальных оценок используются новейшие оценки скорости сходимости в центральной предельной теореме теории вероятностей. В частности, в книге рассматривается ситуация, в которой учитываются возможные изменения надежности восстанавливаемой ИиТС вследствие ее модификаций или ремонтов. Отдельная глава посвящена прямому применению байесовской идеологии при анализе надежности и эффективности ИиТС.

Большая часть книги посвящена учету влияния неоднородности интенсивности потока информативных событий, в результате которых накапливается статистическая информация, на итоговые статистические выводы о параметрах ИиТС. Существенно развита общая теория статистического вывода на основе выборок случайного объема. В книге показано, что при замене объема выборки случайной величиной заметно увеличиваются вероятности критических значений тех или иных статистических критериев или уменьшаются доверительные вероятности по сравнению с классической ситуацией. Детально рассмотрены методы анализа надежности, основанные на предельных теоремах для порядковых статистик в выборках случайного объема. Исследована возможность использования распределения Стьюдента в качестве альтернативы нормальному закону в статистических методах анализа надежности ИиТС.

Книга может быть полезна для специалистов в области применения методов теории вероятностей и математической статистики к анализу надежности ИиТС, а также для аспирантов и студентов старших курсов, обучающихся по специальностям «информатика» и «прикладная математика».

МОНОГРАФИЯ В. Ю. КОРОЛЕВА И И. А. СОКОЛОВА «МАТЕМАТИЧЕСКИЕ МОДЕЛИ НЕОДНОРОДНЫХ ПОТОКОВ ЭКСТРЕМАЛЬНЫХ СОБЫТИЙ» (М.: ТОРУС ПРЕСС, 2008. 192 с.)

Д.ф.-м.н., профессор А. В. Печинкин

В 2008 г. в издательстве «ТОРУС ПРЕСС» вышла монография «Математические модели неоднородных потоков экстремальных событий». Ее авторы — В. Ю. Королёв, профессор, доктор физико-математических наук, профессор факультета вычислительной математики и кибернетики МГУ им. М. В. Ломоносова, ведущий научный сотрудник Института проблем информатики РАН и И. А. Соколов, академик, директор Института проблем информатики РАН.

В книге рассмотрены математические модели вероятностно-статистических характеристик катастроф в неоднородных потоках экстремальных событий. Сформулированы задачи моделирования катастрофических событий, связанных как с критическими превышениями уровня процессом, описывающим накопленные эффекты неблагоприятных факторов, так и с однократными шоковыми воздействиями. В качестве основных математических моделей при решении указанных задач рассматриваются экстремумы обобщенных дважды стохастических пуассоновских процессов и макс-обобщенные дважды стохастические пуассоновские процессы. Для таких процессов доказан ряд предельных теорем. Возникающие в этих теоремах предельные распределения вероятностей предлагаются в качестве аппроксимаций для вероятностно-статистических закономерностей, присущих потокам экстремальных (катастрофических) событий. Большой интерес представляет анализ временных характеристик глобальных катастроф, вызванных столкновением Земли с потенциально опасными небесными телами (астероидами, кометами). На примере этого анализа описаны конкретные процедуры для вычисления вероятностных характеристик катастроф, в частности «ожидаемого времени» катастрофы и продолжительности периода, в течение которого вероятность катастроф пренебрежимо мала.

Книга представляет значительный интерес для специалистов в области применения методов теории вероятностей и математической статистики к анализу рисков, связанных с чрезвычайными ситуациями и катастрофами, и надежности информационных и технических систем. Она также будет полезна аспирантам и студентам старших курсов, обучающимся по специальностям «информатика» и «прикладная математика».

МОНОГРАФИЯ А. И. ЗЕЙФМАНА, В. Е. БЕНИНГА, И. А. СОКОЛОВА «МАРКОВСКИЕ ЦЕПИ И МОДЕЛИ С НЕПРЕРЫВНЫМ ВРЕМЕНЕМ» (М.: ЭЛЕКС-КМ, 2008. 168 с.)

Д.ф.-м.н., профессор С. Я. Шоргин

В 2008 г. в издательстве «ЭЛЕКС-КМ» вышла монография «Марковские цепи и модели с непрерывным временем». Ее авторы — А. И. Зейфман, профессор, доктор физико-математических наук, декан факультета прикладной математики и компьютерных технологий Вологодского государственного педагогического университета, старший научный сотрудник директора Института проблем информатики РАН; В. Е. Бенинг, профессор, доктор физико-математических наук, профессор факультета вычислительной математики и кибернетики МГУ им. М. В. Ломоносова, старший научный сотрудник Института проблем информатики РАН; И. А. Соколов, академик, директор Института проблем информатики РАН.

Как известно, получение явных выражений для вероятностей состояний стохастических моделей возможно лишь в исключительных случаях. В связи с этим одной из важнейших задач при исследовании таких моделей давно является исследование поведения модели при стремлении времени к бесконечности и в частности, скорости сходимости к предельному режиму и связанных с этим функционалов. В рецензируемой книге работе изучаются вопросы, связанные с получением точных оценок скорости к предельному режиму и устойчивости для марковских цепей с непрерывным временем (стационарных и нестационарных), а также приложение методов и результатов к изучению некоторых конкретных моделей, описываемых такими цепями, и в первую очередь, для нестационарных марковских моделей систем массового обслуживания.

Книга, несомненно, вызовет интерес у научных работников, инженеров, аспирантов, студентов и преподавателей вузов, интересующихся современным состоянием исследований в области теории вероятностей и ее приложений.

ABSTRACTS

A DISINTEGRATED PACKET SWITCHING ARCHITECTURE

I. A. Sokolov[1] and V. B. Egorov[2]

[1] IPI RAN, isokolov@ipiran.ru
[2] IPI RAN, vegorov@ipiran.ru

The proposed disintegrated switching architecture enables the designers to create conventional and routing packet switches featuring enlarged performance.

Keywords: packet switch; integrated communication microcontroller; QUICC; PowerQUICC

MEDIAN MODIFICATION OF EM- AND SEM-ALGORITHMS FOR SEPARATION OF MIXTURES OF PROBABILITY DISTRIBUTIONS AND THEIR APPLICATION TO THE DECOMPOSITION OF VOLATILITY OF FINANCIAL TIME SERIES

A. K. Gorshenin[1], V. Yu. Korolev[2], and A. M. Tursunbayev[3]

[1] Department of Mathematical Statistics, Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University, andygorshenin@gmail.com
[2] Department of Mathematical Statistics, Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University, vkorolev@comtv.ru
[3] Department of Mathematical Statistics, Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University

Median modofications of EM- and SEM-algorithms are proposed for separation of mixtures of normal distributions. The advantages of the proposed algorithms over standard methods are illustrated by the numerical solution of the problem of decomposition of volatility of financial indices.

Keywords: separation mixtures of probability distributions; robustness; efficiency; EM-algorithm; SEM-algorithm; volatility

SPLITTING OF DISTRIBUTION MIXTURE IN TWO COMPONENTS

M. P. Krivenko

IPI RAN, mkrivenko@ipiran.ru

The problems of splitting the distribution mixture in two components and of estimating mixture parameters are examined if samples from distribution mixture and from one of components are presented. Two methods of parameters estimation are suggested, as well as corresponding algorithms are constructed and examined.

Keywords: mixture of normal distributions; splitting of distribution mixture; EM-algorithm

CONTINUOUS-TIME NON-HOMOGENEOUS RECURRENT RELIABILITY VARIATION MODELS FOR MODIFIABLE SYSTEMS

S. V. Artyukhov[1] and V. Yu. Korolev[2]

[1]Vneshprombank, ArtyuhovSV@yandex.ru
[2]Department of Mathematical Statistics, Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University; IPI RAN, vkorolev@comtv.ru

Estimates of convergence rate in limit theorems for compound doubly stochastic Poisson processes (compound Cox processes) are used for a more accurate description of the behavior of reliability of complex modifiable technical and information systems within the framework of continuous-time non-homogeneous recurrent reliability variation models

Keywords: compound doubly stochastic Poisson process; compound Cox process; reliability variation model; availability function; guaranteed confidence bounds

INFORMATION TECHNOLOGY OF USING FACIAL BIOMETRICS FOR INCREASING AFIS THROUGHPUT

O. S. Ushmaev

IPI RAN, oushmaev@ipiran.ru

Nowadays, multimodal biometrics is rapidly replacing tedious procedures of identification. Particularly operating and perspective civil ID systems use multimodal approach. The formal method for designing high-speed multibiometric technologies and systems is suggested. The effectiveness of the approach is shown by an example of developed experimental software with service-oriented architecture.

Keywords: biometric identification; multimodal biometrics; platform independent; service-oriented architecture

TIME INTERVALS AS OBJECTS OF A GENERAL-PURPOSE OPERATING SYSTEM

V. Yegorov[1] and E. Matveev[2]

[1]Penza State University; CryptoSoft Corp., vec@cryptosoft.ru
[2]CryptoSoft Corp., eugene@cryptosoft.ru

The main goal of this paper is to describe a new type of the general-purpose operating system object — the time interval object, which allows a general-purpose operating system to act as a real-time operating system.

Keywords: operating system; real time; time interval; extension of C language; hardware interrupt controller; multiprocessor system

EUROWORDNET: OBJECTIVES, STRUCTURE, AND RELATIONSHIPS

O. Kozhunova

[1]IPI RAN, okozhunova@ipiran.ru

In the review, brief desciption of the EuroWordNet tool, history of its creation, examples of the similar resources, its objectives, structure, and relationships are given.

Keywords: lexical and semantic resource EuroWordNet; WordNet dictionaries; thesaurus; sinset; language relationships; OnterLingual Index ILI

Об авторах

Артюхов Сергей Владимирович (р. 1983) — заместитель начальника аналитического отдела ООО «ВНЕШПРОМБАНК»

Горшенин Андрей Константинович (р. 1986) — аспирант кафедры математической статистики факультета вычислительной математики и кибернетики МГУ им. М. В. Ломоносова

Егоров Валерий Юрьевич (р. 1974) — кандидат технических наук, обучается в докторантуре Пензенского государственного университета, доцент кафедры «Вычислительная техника» Пензенского государственного университета

Егоров Владимир Борисович (р. 1948) — кандидат технических наук, и.о. ведущего научного сотрудника ИПИ РАН

Кожунова Ольга Сергеевна (р. 1982) — научный сотрудник ИПИ РАН

Королёв Виктор Юрьевич (р. 1954) — доктор физико-математических наук, профессор кафедры математической статистики факультета вычислительной математики и кибернетики МГУ им. М. В. Ломоносова, ведущий научный сотрудник ИПИ РАН

Кривенко Михаил Петрович (р. 1946) — доктор технических наук, профессор, ведущий научный сотрудник ИПИ РАН

Матвеев Евгений Анатольевич (р. 1961) — директор ООО НТП «Криптософт»

Соколов Игорь Анатольевич (р. 1954) — академик (действительный член) Российской академии наук, доктор технических наук, директор ИПИ РАН

Турсунбаев Асхат Маратович (р. 1985) — выпускник кафедры математической статистики факультета вычислительной математики и кибернетики МГУ им. М. В. Ломоносова

Ушмаев Олег Станиславович (р. 1981) — кандидат технических наук, старший научный сотрудник ИПИ РАН

About Authors

Artuykhov Sergei V. (b. 1983) — deputy head of the analytical department, Vneshprombank

Egorov Vladimir B. (b. 1948) — Candidate of Technical Sciences (PhD), leading scientist, Institute of Informatics Problems, Russian Academy of Sciences

Gorshenin Andrey K. (b. 1986) — postgraduate student, Department of Mathematical Statistics, Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University

Korolev Victor Yu. (b. 1954) — Doctor of Science in physics and mathematics; professor, Department of Mathematical Statistics, Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University; leading scientist, Institute of Informatics Problems, Russian Academy of Sciences

Kozhunova Olga S. (b. 1982) — scientist, Institute of Informatics Problems, Russian Academy of Sciences

Krivenko M. P. (b. 1946) — Doctor of Science; professor; leading scientist, Institute of Informatics Problems, Russian Academy of Sciences

Matveev Eugene A. (b. 1961) — director, CryptoSoft Corp.

Sokolov Igor A. (b. 1954) — Academician of the Russian Academy of Sciences; Doctor of Technical Sciences; director, Institute of Informatics Problems, Russian Academy of Sciences

Tursunbayev Askhat M. (b. 1985) — graduate, Department of Mathematical Statistics, Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University

Ushmaev Oleg S. (b. 1981) — Candidate of Technical Sciences (PhD), senior scientist, Institute of Informatics Problems, Russian Academy of Sciences

Yegorov Valery Yu. (b. 1974) — Candidate of Technical Sciences (PhD), senior developer, CryptoSoft Corp.; associate professor, Computer Science Department of Penza State University

АВТОРСКИЙ УКАЗАТЕЛЬ ЗА 2008 г.

2008 AUTHOR INDEX

Правила подготовки рукописей статей для публикации в журнале «Информатика и её применения»

Журнал «Информатика и её применения» публикует теоретические, обзорные и дискуссионные статьи, посвященные научным исследованиям и разработкам в области информатики и ее приложений. Журнал издается на русском языке. Тематика журнала охватывает следующие направления:

- теоретические основы информатики;
- математические методы исследования сложных систем и процессов;
- информационные системы и сети;
- информационные технологии;
- архитектура и программное обеспечение вычислительных комплексов и сетей.

1. В журнале печатаются результаты, ранее не опубликованные и не предназначенные к одновременной публикации в других изданиях. Публикация не должна нарушать закон об авторских правах. Направляя свою рукопись в редакцию, авторы автоматически передают учредителям и редколлегии неисключительные права на издание данной статьи на русском языке и на ее распространение в России и за рубежом. При этом за авторами сохраняются все права как собственников данной рукописи. В связи с этим авторами должно быть представлено в редакцию письмо в следующей форме: Соглашение о передаче права на публикацию:

   «Мы, нижеподписавшиеся, авторы рукописи « », передаем учредителям и редколлегии журнала «Информатика и её применения» неисключительное право опубликовать данную рукопись статьи на русском языке как в печатной, так и в электронной версиях журнала. Мы подтверждаем, что данная публикация не нарушает авторского права других лиц или организаций. Подписи авторов: (ф. и. о., дата, адрес)».

   Редколлегия вправе запросить у авторов экспертное заключение о возможности опубликования представленной статьи в открытой печати.
2. Статья подписывается всеми авторами. На отдельном листе представляются данные автора (или всех авторов): фамилия, полные имя и отчество, телефон, факс, e-mail, почтовый адрес. Если работа выполнена несколькими авторами, указывается фамилия одного из них, ответственного за переписку с редакцией.
3. Редакция журнала осуществляет самостоятельную экспертизу присланных статей. Возвращение рукописи на доработку не означает, что статья уже принята к печати. Доработанный вариант с ответом на замечания рецензента необходимо прислать в редакцию.
4. Решение редакционной коллегии о принятии статьи к печати или ее отклонении сообщается авторам. Редколлегия не обязуется направлять рецензию авторам отклоненной статьи.
5. Корректура статей высылается авторам для просмотра. Редакция просит авторов присылать свои замечания в кратчайшие сроки.
6. При подготовке рукописи в MS Word рекомендуется использовать следующие настройки. Параметры страницы: формат — A4; ориентация — книжная; поля (см): внутри — 2,5, снаружи — 1,5, сверху — 2, снизу — 2, от края до нижнего колонтитула — 1,3. Основной текст: стиль — «Обычный»: шрифт Times New Roman, размер 14 пунктов, абзацный отступ — 0,5 см, 1,5 интервала, выравнивание — по ширине. Рекомендуемый объем рукописи — не свыше 25 страниц указанного формата. Ознакомиться с шаблонами, содержащими примеры оформления, можно по адресу в Интернете: http://www.ipiran.ru/journal/template.doc.
7. К рукописи, предоставляемой в 2-х экземплярах, обязательно прилагается электронная версия статьи (как правило, в форматах MS WORD (.doc) или LaTex (.tex), а также — дополнительно — в формате .pdf) на дискете, лазерном диске или по электронной почте. Сокращения слов, кроме стандартных, не применяются. Все страницы рукописи должны быть пронумерованы.
8. Статья должна содержать следующую информацию на русском и английском языках: название, Ф.И.О. авторов, места работы авторов и их электронные адреса, аннотация (не более 100 слов), ключевые слова. Ссылки на литературу в тексте статьи нумеруются (в квадратных скобках) и располагаются в порядке их первого упоминания. Все фамилии авторов, заглавия статей, названия книг, конференций и т. п. даются на языке оригинала, если этот язык использует кириллический или латинский алфавит.
9. Присланные в редакцию материалы авторам не возвращаются.
10. При отправке файлов по электронной почте просим придерживаться следующих правил:
    - указывать в поле subject (тема) название журнала и фамилию автора;
    - использовать attach (присоединение);
    - в случае больших объемов информации возможно использование общеизвестных архиваторов (ZIP, RAR);
    - в состав электронной версии статьи должны входить: файл, содержащий текст статьи, и файл(ы), содержащий(е) иллюстрации.
11. Журнал «Информатика и её применения» является некоммерческим изданием, и гонорар авторам не выплачивается.

Адрес редакции: Москва 119333, ул. Вавилова, д. 44, корп. 2, ИПИ РАН
Тел.: +7 (499) 135-86-92 Факс: +7 (495) 930-45-05 E-mail: rust@ipiran.ru