Раздел 4. Теоретические основы построения информационных технологий для интеллектуальных систем автоматизации

Многоагентная технология принятия решений в задачах объединения данных[1]

В. И. Городецкий, О. В. Карсаев, В. В. Самойлов

Санкт-Петербургский институт информатики и автоматизации РАН
{gor, ok, samovl}@mail.iias.spb.su
http://space.iias.spb.su/ai/

УДК 681.3
В. И. Городецкий, О. В. Карсаев, В. В. Самойлов. **Многоагентная технология принятия решений в задачах объединения данных** // Труды СПИИРАН. Вып. 1, т. 2. — СПб.: СПИИРАН, 2002.

Аннотация. *В статье рассматривается технология поддержки принятия решений на основе информации, полученной из различных источников. Специфика технологии разработки таких систем обусловлена распределенностью источников и гетерогенностью представленных в них данных, распределенным характером функционирования таких систем поддержки принятия решений, а также распределенным характером процесса разработки ее базовых компонент. Эти особенности ведут к необходимости решения ряда специфических задач технологического характера. Модели, методы и архитектурные решения, касающиеся технологии разработки систем поддержки принятия решений на основе распределенных гетерогенных источников данных, а также их реализация в форме программной инструментальной системы составляют основное содержание данной работы.* — Библ. 46 назв.

UDK 681.3.
V. I. Gorodetski, O. V. Karsayev, and V. V. Samoilov. **Multi-agent data fusion technology** // SPIIRAS Proceedings. Issue 1, v. 2. — SPb: SPIIRAS, 2002.

Abstract. *The paper is devoted to the problems associated with the development of data fusion systems. Peculiarities of any data fusion development technology are caused several reasons. The most distributed mode of data fusion system operation and also distributed character of the development technology itself. These peculiarities entail a need to solve a number of very specific technology-oriented tasks. Models, methods and architectural decisions associated with the above development technology and also technology support mean implemented as a specialized software tool are the focus of the paper.* — Bibl. 46 items.

1. Введение

Задачи слияния данных или слияния информации относятся к области совместной обработки данных, которые получаются из различных распределенных источников и которые характеризуют одну и ту же сложную ситуацию, состояние объекта и т.п. В локальных источниках могут содержаться данные раз-

[1] Работа поддерживается Российским фондом фундаментальных исследований (грант № 01-01-00109а)

личной физической природы, различной точности и надежности, эти данные могут обладать различной степенью полноты, могут быть представлены в различных структурах данных и т.д. Такие задачи возникают в различных прикладных областях, например, при мониторинге и прогнозировании состояния окружающей среды и природных катастроф, оценке состояний сложных объектов и управлении ими (например, атомных электростанций, электрических сетей), оценке и прогнозировании ситуаций (например, в системе надзора за некоторой зоной на море, в воздушном пространстве и т.п.), и в других приложениях. Цель задачи слияния данных — это принятие решений по множеству фрагментов распределенных данных.

Начало этому направлению положили исследования, проводившиеся в интересах Министерства обороны США в середине 80-х годов. Эти исследования касались совместной обработки информации военного и экономического характера, полученной о некотором объекте или о его состоянии с помощью различных средств космической и аэрофотосъемки, например, фото и телевизионной съемки в видимом диапазоне волн электромагнитного спектра, в инфракрасном диапазоне, в других диапазонах спектра (многоспектральные данные), полученные при различных положениях солнца, в различное время года, и т.д. Хотя и сейчас значительные усилия тратятся здесь на исследования в военной области, появилось много иных приложений, стимулирующих постоянное возрастание интереса к данным исследованиям.

До последнего времени было известно совсем немного публикаций о прикладных задачах, относящихся к данной области (см. например, в [13] описание приложения из области сенсорных сетей, в [27] — задачу мониторинга сейсмической активности). В последние несколько лет интерес к методам и моделям слияния данных со стороны различного рода приложений неизмеримо возрос. Приведем два примера таких задач.

1. Обнаружение вторжений в компьютерную сеть [5]. Основную угрозу для компьютерных сетей представляют распределенные атаки, выполняемые группой злоумышленников по некоторому сценарию. "Следы" атак проявляются в различных данных, порождаемых системой защиты сети. Например, они проявляются в данных *tcpdump*, получаемых предобработкой сетевого трафика, в данных аудита, в последовательности системных вызовов операционной системы, в данных мониторинга работы серверов приложений, обращений к базам данных, директориям, в данных анализа профилей пользователей и т.д. Такие данные порождаются на различных компьютерах сети. Своевременное обнаружение нелегитимной деятельности пользователей и атак возможно только при совместном анализе всех данных. Формально, задача обнаружение атак относится к задачам классификации деятельности пользователей, которая решается на основе многоуровневого объединения данных и решений множества распределенных средств защиты сети, ведущих мониторинг и анализ трафика и данных, порождаемых различными программами системы защиты.

2. Анализ и прогноз развития опасных ситуаций. В различных регионах ежедневно возникают аварийные и опасные ситуации. Они могут возникать как следствие природных катастроф (землетрясений, наводнений, и т.д.), техногенных катастроф на производствах (химическое, ядерное), и т.п. Особенностью таких явлений является быстрое развитие и пространственное распространение, зависимость от погоды (направления и силы ветра, температуры и т.д.), ландшафта, инфраструктуры зданий и т.п. Для оценки состояния и прогноза развития подобных ситуаций необходимо использовать информацию из раз-

Труды СПИИРАН. 2002. Вып. 1. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2002. Issue 1. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

личных источников. Локальными источниками данных могут быть данные сенсоров, метеорологические данные, информация со спутников, сообщения людей, математические модели и т.д.

Задачи описанного типа принято кратко называть *задачами слияния данных* ("*data fusion*")

К настоящему времени предложены различные подходы к принятию решений в задачах слияния данных [19]. В одном из них в процесс принятия решений вовлекаются сразу все данные локальных источников, так что любой локальный решатель (классификатор), а их, как правило, требуется несколько, может использовать данные из различных источников. В другом подходе строится иерархия классификаторов, в которой сначала принимаются решения по данным локальных источников, а потом эти решения объединяются для получения итогового решения. Существует и третий вариант, когда при комбинировании решений локальных источников используются также и некоторые данные [10]. В данной работе в качестве базового рассматривается второй из трех названных подходов, поскольку в большинстве прикладных задач слияния данных он наиболее перспективен.

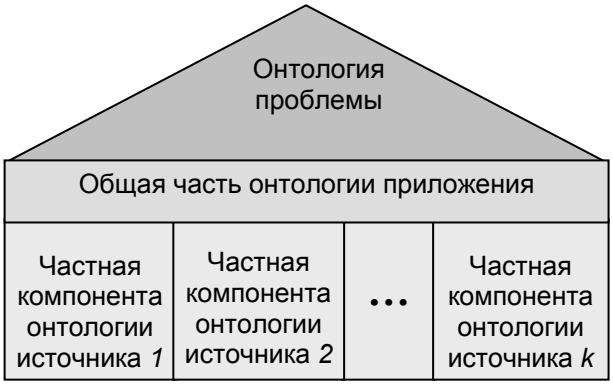

Рис.1. Структура распределенной онтологии

В рамках данной статьи рассматриваются специфические вопросы, связанные с моделями, методами и технологией разработки распределенных систем слияния данных, построенных на основе многоагентной архитектуры. В частности, в разделе 2 обосновывается необходимость построения мета-модели источников данных, а также рассматриваются специфические проблемы и наиболее важные компоненты технологии ее реализации. В разделе 3 рассматривается мета-модель объединения решений, сформированных на основе данных различных источников. В разделе 4 дается описание многоагентной архитектуры инструментальной системы, предназначенной для разработки приложений в области слияния данных. В заключении формулируются основные выводы и приводятся сведения о состоянии работ по созданию программного инструментария для создания приложений в области систем слияния данных.

2. Мета-модель данных

2.1 Особенности данных распределенных источников

В рассматриваемой задаче данные являются *распределенными* и *гетерогенными*, т.е. они представлены в распределенных базах данных и характеризуются разнообразием физической природы, шкал измерения и структур представления, различием степеней полноты и неопределенности и т.д. Распределенность и гетерогенность данных создают ряд специфических проблем при разработке приложений, которые нехарактерны для других классов задач принятия решений. Первая из них — это проблема обеспечения *глобальной однозначности семантики* терминов, используемых при спецификации данных локальных источников. Эта проблема возникает из-за того, что спецификация

Труды СПИИРАН. 2002. Вып. 1. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2002. Issue 1. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

данных выполняется различными распределенными пользователями. Они могут использовать одинаковые термины в различном смысле и, наоборот, для одного и того же понятия могут использовать различные названия. В соответствии с современными взглядами, для решения проблемы однозначного понимания терминов необходимо использование высокоуровневой модели знаний, разделяемой всеми сущностями системы, т.е. согласованной и доступной каждой из них.

Обычно эта база знаний строится в терминах онтологии приложения. Предполагается, что именно этот подход используется в нашем случае. Обобщенная структура онтологии некоторого приложения проблемы слияния данных представлена на рис. 1.

Вторая проблема известна как *проблема идентификации экземпляров сущностей.* Она возникает в связи с тем, что информация об одной и той же сущности (ситуации, состоянии объекта и т.п.) представляется в распределенной форме, а потому необходимо иметь специальные механизмы для отождествления компонент данных и сущности, описание которой они представляют. Эта проблема должна решаться для того, чтобы можно было *выбирать* и *анализировать совместно* информацию об одной и той же сущности. Заметим, что при этом информация о некоторых сущностях в отдельных источниках может отсутствовать. Графическое пояснение к проблеме идентификации сущностей дается на рис. 2. На нем примеры, относящиеся к описанию одной и той же ситуации, но представленные в различных источниках данных, имеют одинаковую нумерацию и соединены пунктирными линиями. Заметим, что не все примеры имеют соответствующую компоненту описания во всех источниках данных. Например, описание ситуации 9 представлено только в двух источниках данных, а ситуации 2, 3, 7 и др. вообще представлены только в одном из источников.

Одной из особенностей формирования онтологии в данной задаче по сравнению с аналогичными задачами, которые возникают в других классах приложений, состоит в том, что здесь данные представлены на ином языке, чем онтология (на языке соответствующей СУБД, а запросы к ним должны формулироваться на языке SQL). Данные представлены на языке соответствующей СУБД, а запросы к ним должны формулироваться на языке SQL, а компоненты онтологии описываются на одном из языков типа XML, RDF или DAML+OIL, поэтому возникает серьезная проблема переформулирования запросов с языка спецификации онтологии на язык SQL.

Наконец, еще одна проблема возникает в связи с тем, что множества атрибутов различных источников могут пересекаться, и при этом *одинаковые* или "*сходные*" свойства в различных источниках могут быть *представлены в различных структурах и шкалах измерения* (номинальной, числовой и т.д., могут быть представлены в форме изображений, сигналов, экспертных данных), или просто представлены с различной точностью.

Существуют и другие особенности, однако перечисленные являются основными.

Вышеупомянутые особенности данных распределенных источников обуславливают специфику технологии формирования онтологии приложений проблемы слияния данных, которая оказывает существенное влияние на архитектуру системы, возможный выбор методов и алгоритмов и технологию разработки. Основные аспекты этой технологии рассматриваются в следующем подразделе.

Труды СПИИРАН. 2002. Вып. 1. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2002. Issue 1. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

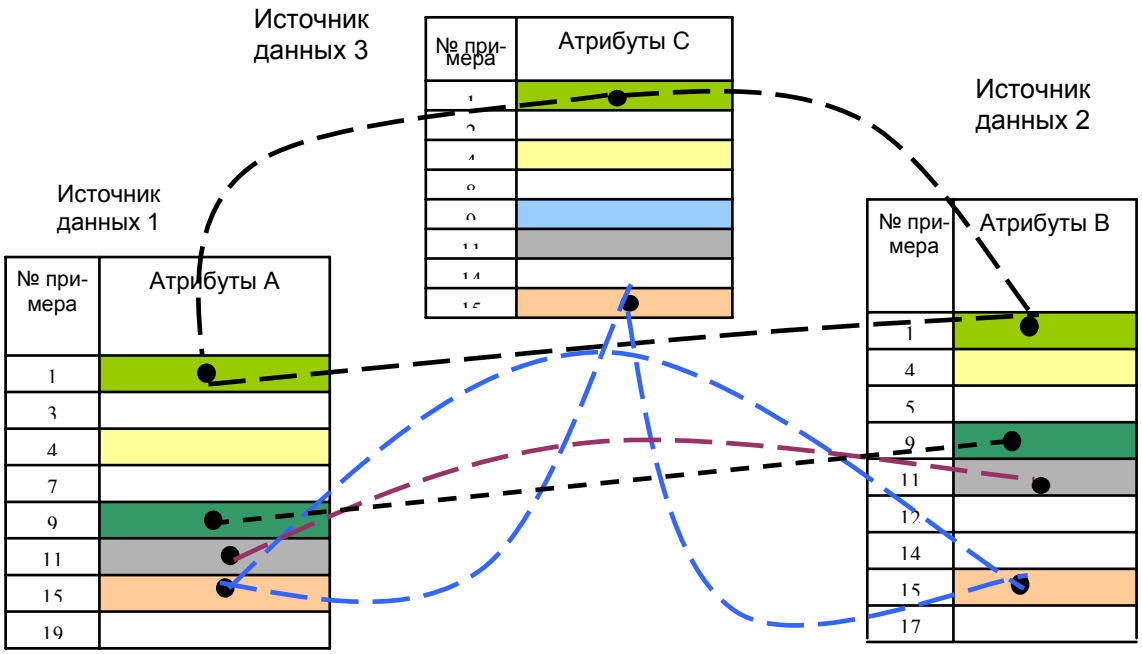

Рис.2. Пояснение проблемы идентификации сущностей

2.2. Технология разработки распределенной онтологии и мета-модели данных

Мета-модель данных — это представление распределенных гетерогенных данных на мета-уровне, поддерживающая единый взгляд на данные в разных источниках. Мета-модель данных является составной частью онтологии, поэтому разработка мета-модели данных выполняется в процессе разработки последней. В предложенной технологии она разрабатывается распределенным коллективом менеджеров источников данных под руководством менеджера мета-данных. Общая схема взаимодействия распределенного коллектива разработчиков онтологии представлена на рис. 3. В этой схеме предполагается, что каждому менеджеру данных ассистирует агент, который поддерживает взаимодействие "своего" менеджера с менеджером мета-данных.

В реализуемой программно схеме разработки онтологии предполагается, что на первом шаге менеджеры источников данных, пользуясь проблемной онтологией (она одинакова для всех приложений), разрабатывают компоненты онтологии, которые описывают "свой" источник данных. Здесь эта процедура не рассматривается, поскольку она является отдельной научной и технической проблемой, которая не относится к специфическим проблемам задачи слияния данных. Подробная информация по этой проблеме может быть почерпнута из [38]. Здесь рассмотрим только основные идеи решения специфических проблем, о которых шла речь в предыдущем подразделе.

Проблема обеспечения *глобальной однозначности семантики* терминов, используемых в онтологии приложения, решается на основе специального протокола переговоров агентов, ответственных за формирование компонент онтологии приложения. На рис.1 такими агентами являются агенты управления данными источников и агент мета-уровня, называемый KDD-мастером. Пример протокола распределенной разработки онтологии приложения (он включает в себя в качестве компоненты еще один протокол, названный "протоколом син-

Труды СПИИРАН. 2002. Вып. 1. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2002. Issue 1. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

хронизации"), решающего проблему однозначности семантики терминологии, представлен на рис. 4. Он предполагает, что базовый вариант терминологии предлагается менеджером мета-данных, и этот вариант далее модифицируется в процессе итераций, которые выполняются агентами на основе специального протокола синхронизации. Этот протокол здесь не рассматривается ввиду ограниченности объема статьи.

Вторая проблема — проблема идентификации экземпляров сущностей (см. иллюстрацию проблемы на рис. 2), решается с помощью специального алгоритма, суть которого кратко состоит в следующем. Для каждого экземпляра сущности онтологии (речь идет о сущности, относительно состояний которой

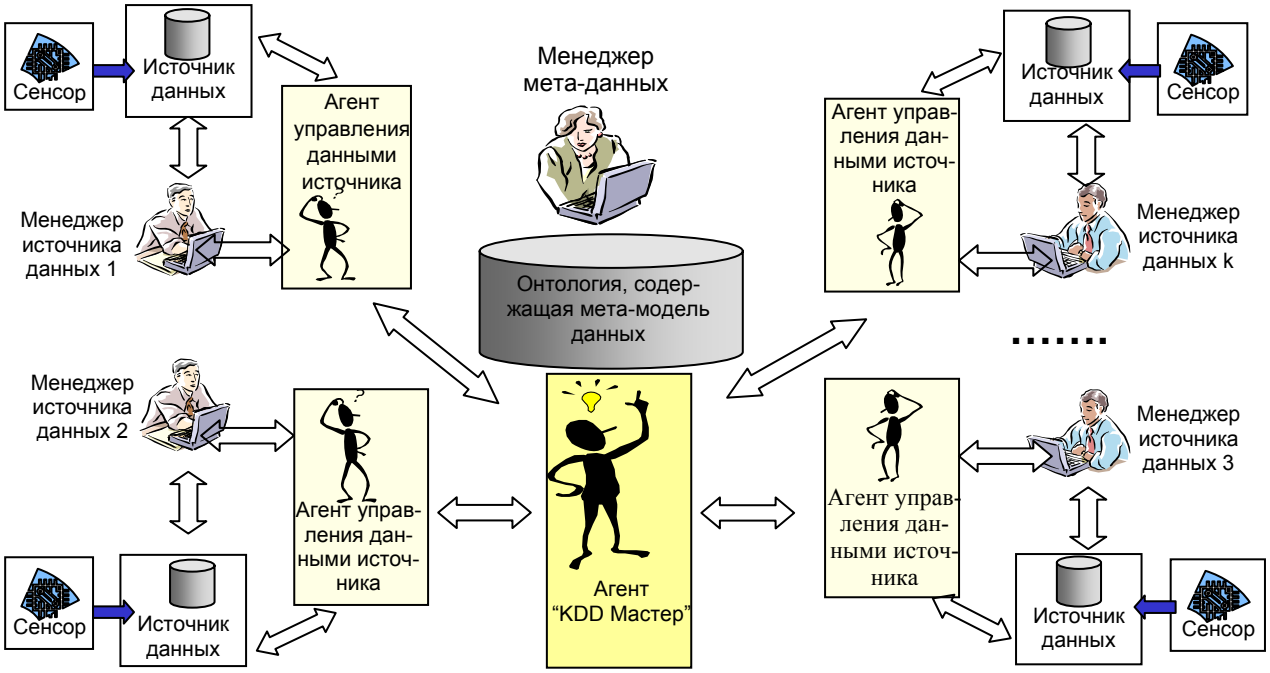

Рис.3. Схема взаимодействия менеджеров источников данных и менеджера мета-данных при разработке распределенной онтологии

должно приниматься решение) вводится понятие "*Идентификатор сущности*" ("*ID-entity*"). Этот идентификатор далее играет роль первичного ключа экземпляра (по аналогии с первичным ключом таблицы базы данных). Для каждого такого идентификатора в общей части онтологии приложения вводится такое правило (функция), которое может быть использовано для вычисления ключа экземпляра объекта. Это правило строится как функция выбранного подмножества атрибутов объекта. Такое правило определяется для каждого источника данных, определяя локальный первичный ключ.

Что касается проблемы обеспечения совместимости запросов к базам данных, формулируемых в терминах языка онтологии (RDF, DAML+OIL или другого языка), то эта проблема в рамках разрабатываемой технологии решается с помощью использования виртуальных объектов (в базах данных такие объекты имеют тип *VIEW*). Заметим, что принятый в настоящее время подход, в котором понятия и отношения онтологии, а также экземпляры этих понятий, описываются с помощью одного и того же языка [25], в нашем случае неприемлем, по-

Труды СПИИРАН. 2002. Вып. 1. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2002. Issue 1. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

скольку в рассматриваемом классе приложений эти экземпляры обязательно хранятся в базах данных.

Рассмотрим кратко путь решения проблемы различного представления одних и тех же атрибутов в различных источниках данных. Пусть *X* один из таких атрибутов. В общей части онтологии приложения эксперт должен определить тип этого атрибута и шкалу его измерения. Этот процесс выполняется путем переговоров с менеджерами источников данных на основе специального протокола. Для всех источников, где атрибут *X* присутствует, определяется выражение, которое преобразует его представление в источнике к типу и шкале, которые приняты на основе переговоров. Далее, каждый раз, когда данный атрибут является компонентой запроса к соответствующей базе данных, он конвертируется в представление, принятое на мета-уровне и единое для всех источников.

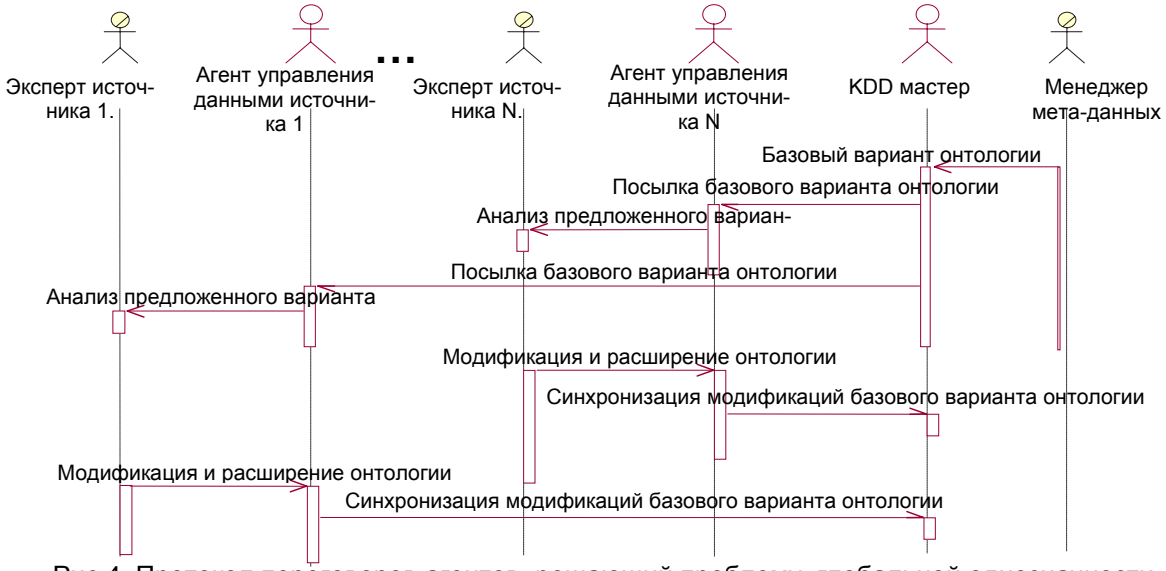

Рис.4. Протокол переговоров агентов, решающий проблему глобальной однозначности семантики терминов онтологии.

В результате перечисленных выше операций получается структурное представление онтологии, изображенное на рис. 5. В этой структуре явно указывается, что наряду с традиционными компонентами онтологии, онтология приложения проблемы слияния данных содержит дополнительный набор функциональностей, поддерживающих разработку, модификацию и использование онтологии.

3. Мета-модель и методы объединения решений

В мета-модели объединения решений следует выделить два самостоятельных аспекта. Первый из них касается спецификации многоуровневой схемы принятия решений. В данной работе рассматривается только задача классификации, т.е. принимается предположение, что результатом слияния данных является решение о классе состояний некоторого объекта, информация о котором получается из различных источников. Второй аспект касается мета-модели данных, которые используются для обучения и тестирования решателей (в нашем случае — классификаторов). Назначение этой модели состоит в том, чтобы произвести распределение этих данных между различными классификаторами и уровнями принятия решений. Рассмотрим кратко обе названные модели.

Труды СПИИРАН. 2002. Вып. 1. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2002. Issue 1. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

3.1. Схемы объединения решений

В данном подразделе рассматриваются варианты схем объединения решений классификаторов локальных источников в общей схеме слияния данных. Эти схемы далее положены в основу многоагентной архитектуры соответствующей программной системы. Рассмотрим различные варианты таких схем, которые реально могут встретиться в задачах слияния данных. Выбор схемы в конкретном приложении оказывает существенное влияние на выбор архитектуры реализующей ее программной системы. В самом простом случае, когда размерность вектора атрибутов локального источника достаточно мала (в пределах 20-30) и данные достаточно "хорошие" (например, либо только числовые, либо только дискретные), может оказаться достаточным (с позиций качества формируемых решений и вычислительной эффективности) использовать всего один классификатор базового уровня, решение которого передается на мета-уровень. Этому случаю соответствует рис.6.

На рис.7 представлен случай, когда решения всех классификаторов базового уровня передаются на мета-уровень, где они объединяются с решениями классификаторов, "работающих" с остальными источниками данных. На рис.8 представлен случай, когда на локальном источнике данных имеется мета-классификатор, который объединяет решения лишь некоторых классификаторов базового уровня. Заметим, что таких мета-классификаторов на отдельном источнике может быть и несколько. На мета-уровень принятия решений в этом случае передаются как решения мета-классификаторов, так и решения некоторых классификаторов базового уровня. Рис.9 отвечает случаю, когда на уровень объединения решений, полученных на разных источниках, передаются только решения мета-классификаторов. В самом общем случае возможен вариант, когда уже на уровне отдельного источника данных имеются два или бо-

Рис.6. Простейший случай: Один классификатор базового уровня на источнике данных

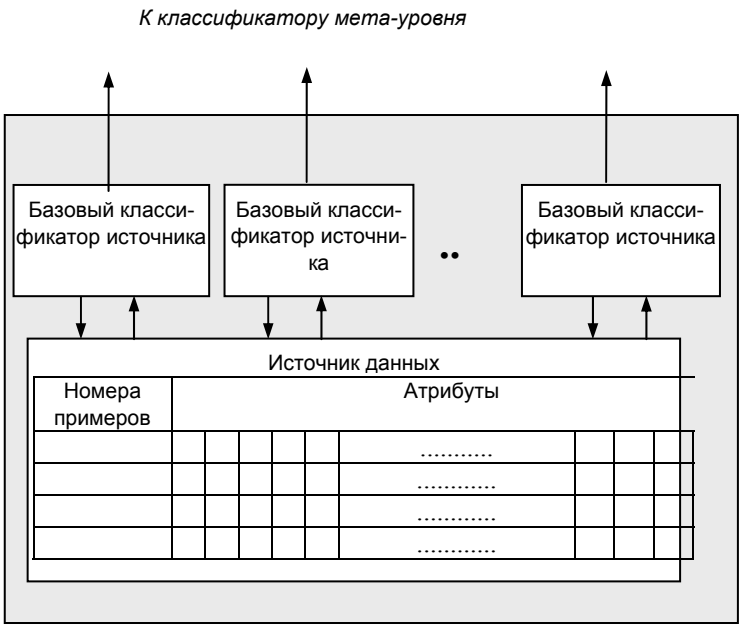

Рис.7. Комбинирование решений: Вариант 1

Труды СПИИРАН. 2002. Вып. 1. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2002. Issue 1. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

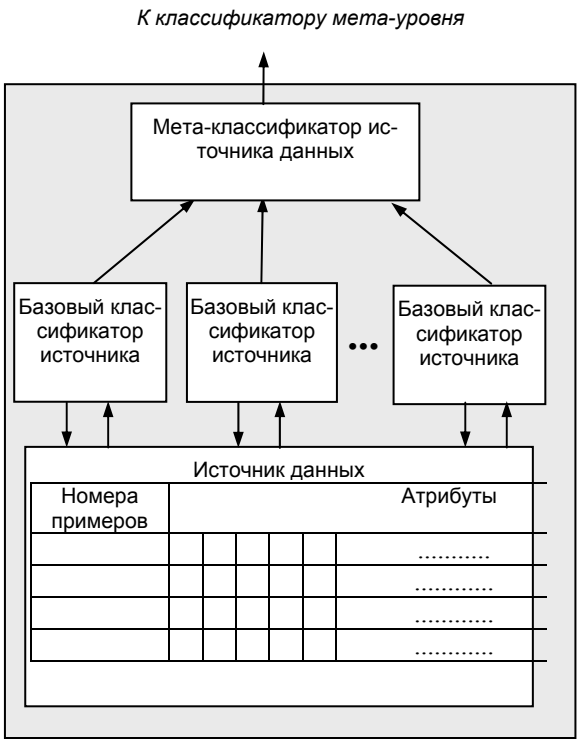

Рис.8. Комбинирование решений: вариант 2

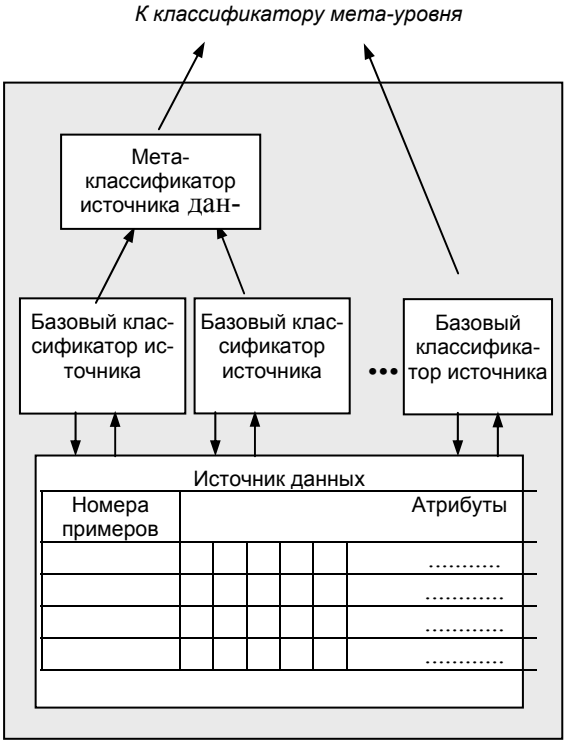

Рис.9. Комбинирование решений: вариант 3

лее уровня обобщения. Очевидно, что это возможно тогда, когда данные источника характеризуются большим количеством весьма разнообразных атрибутов (по структуре представления, по точности и т.д.), а число записей в базе данных для обучения классификаторов[1] очень велико.

На мета-уровне объединяются решения, которые представляют собой решения классификаторов источников данных. По этой причине все они измерены либо в булевой шкале, либо в номинальной (тип шкалы представления зависит от принятого метода объединения данных и от количества классов). Но по структуре схемы объединения данных полностью повторяют варианты, представленные на рис. 6–9 с тем лишь отличием, что во всех случаях приходится иметь дело с классификаторами мета-уровня, хотя это отличие сводится лишь к типу шкал атрибутов, о чем только что шла речь.

2.3. Мета-модель распределенных данных для обучения и тестирования

В известных подходах классификаторы базового уровня должны быть различными. Чтобы быть таковыми, они должны различаться хотя бы в одном из двух измерений, а именно, они должны быть либо построены на основе разных алгоритмов, либо должны использовать различные данные для обучения. Однако в случае распределенного обучения, что типично для задач слияния данных, необходимо принимать во внимание еще и третье измерение, а именно, различие в множествах атрибутов, значения которых могут быть использованы для принятия решений в каждом из источников.

[1] Речь идет о базе обучающих данных для одной и той же базовой сущности.

Труды СПИИРАН. 2002. Вып. 1. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2002. Issue 1. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

Этот факт поясняется на рисунках 10, 11[1]. На них схематически показаны два источника данных и распределение этих данных для обучения базовых классификаторов. Эти данные расщеплены на несколько отдельных наборов (на рисунках показаны только два из них), имитирующие различные источники. Они также расщеплены в пределах каждого из двух показанных источников в соответствии с их использованием для обучения базовых классификаторов.

В первом источнике данных (рис.10) все 3 классификатора базового уровня строятся для одного и того же набора атрибутов, однако при обучении и тестировании они используют различные наборы данных, а также могут быть построены на базе различных алгоритмов обучения. Кроме того, в этой схеме выделена часть данных, которая не вовлекается в процесс обучения классификаторов, а предназначена для формирования мета-данных или обучения и тестирования рефери (см. подраздел 3.3).

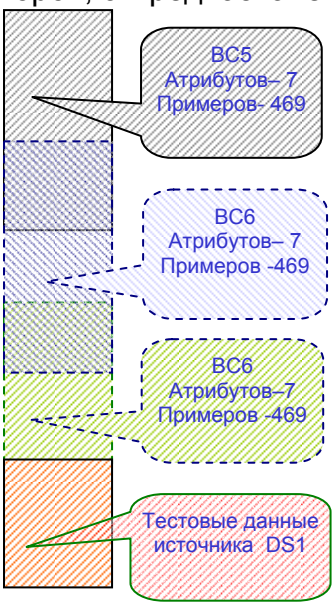

Рис.10. Источник данных 1

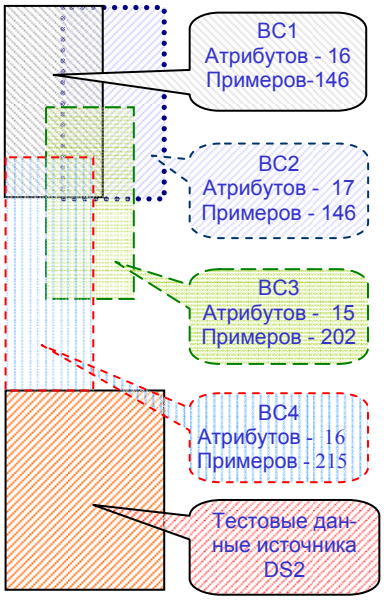

Рис. 11. Источник данных 2

Разбиение обучающих данных второго источника (рис. 11) демонстрирует более общий случай. В нем классификаторы базового уровня различаются по всем трем измерениям.

Приведенный пример свидетельствует о необходимости специальной модели и процедур управления распределенными данными. Управление распределенными данными имеет целью распределение тренировочных и тестовых данных для обучения базовых классификаторов, а также управление вычислением мета-данных для обучения классификаторов верхних уровней; В архитектуре эта задача возлагается на специальных агентов, работающих как с данными локальных источников, так и с данными мета-уровня. (см. раздел 4).

Названные задачи решаются с использованием специального протокола переговоров, в которых принимают также участие менеджеры источников данных и менеджер мета-уровня.

3.3. Краткий обзор известных схем объединения решений классификаторов

К настоящему времени предложено много различных методов комбинирования решений классификаторов *базового уровня* решающих одну и ту же задачу. Эти методы можно условно разделить на три группы:

1. Методы, использующие в той или иной форме голосование.

2. Методы типа мета-классификации, основанные на использовании мета-данных. В зарубежной литературе такие методы объединяются термином "stacked generalization" [46].

[1] В примере рис. 10, 11 использованы данные, которые были предложены на IEEE соревнованиях программ извлечения знаний из данных [24] в 1999 году.

Труды СПИИРАН. 2002. Вып. 1. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2002. Issue 1. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

3. Методы, использующие оценку компетентности классификаторов.

4. Методы, использующие комбинирование свидетельств Демстера-Шафера.

Рассмотрим кратко суть этих методов, кроме последнего. Он не рассматривается ввиду того, что авторам данной работы он представляется неперспективным.

Методы комбинирования решений классификаторов на основе *голосования* возникли исторически первыми [3, 12]. Благодаря своей простоте и, в основном, удовлетворительной точности они активно используются и в настоящее время. Самый простой из них был предложен в работе [12]. Он известен под названием метода простого голосования ("unweighted vote", или "majority vote") [15, 16]). Если результатом работы множества классификаторов являются условные вероятности классов для заданного вектора атрибутов, то простое суммирование вероятностей в пользу каждого из классов с последующей нормировкой результатов практически реализует ту же схему голосования, но в этом случае уже можно говорить о взвешенном голосовании ("weighted vote"). Разработано также много других методов взвешенного голосования [35, 23, 9, 4, 24, etc.].

Однако в настоящее время наибольшим вниманием исследователей и прикладников пользуются методы, использующие в той или иной форме знания о свойствах классификаторов базового уровня [36]. Все они в разной степени и в разной форме используют мета-данные и мета-обучение.

Исторически методы этой группы зародились из идеи "обобщения по типу матрешки" ("*stacked generalization*"). Эта идея для более общего случая, чем обучение классификаторов, была предложена в работе [46]. Сама идея "stacked generalization" очень проста, однако она оказалась весьма продуктивной во многих приложениях, в том числе, и для построения методов мета-обучения. Применительно к задаче объединения решений классификаторов идея "stacked generalization" может быть пояснена с помощью алгоритма обучения классификации, предложенного в работе [42]. Суть метода поясняется на рис.12.

Пусть имеется некоторое множество обучающих данных ***L*** и выбрано некоторое множество алгоритмов $A_1,...,A_K$ [1], обучение которых приводит к получению *K* разных классификаторов. Множество таких классификаторов названо выше классификаторами базового уровня. Иногда их называют также классификаторами уровня *0*. По результатам тестирования (оно проводится, например, по схеме известной под названием *cross-validation* или на новых данных) строится множество мета-данных. Это множество состоит из кортежа меток классификаций, которые являются выходами алгоритмов $A_1,...,A_K$, которое дополнено меткой правильной классификации. Таким образом, мета-данные представляются множеством кортежей вида $(z_{1i},\ldots,z_{Ki},\ y_i)$, где *I* — номер примера из тестовых данных, *1,...,K* — номера классификаторов, z_{ki} — решение алгоритма A_k о классе принадлежности примера с номером *i*, y_i — метка правильной классификации этого же примера. Далее мета-данные используются для обучения классификатора мета-уровня, называемого иногда также мета-классификатором.

[1] Лучше, если эти алгоритмы различные. Заметим, что это требование полезно использовать в любом методе комбинирования решений.

Труды СПИИРАН. 2002. Вып. 1. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2002. Issue 1. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

Частным случаем такого подхода является схема мета-классификации[1] [10, 36], разработанная независимо от работы [46] специально для распределенного обучения и параллельной реализации. Этот метод допускает использование распределенной базы обучающих данных, для каждой их которых строится несколько классификаторов, работающих параллельно. Эти классификаторы, называемые здесь "базовыми", тестируются на новых данных. Результаты тестирования используются как мета-данные, которые, в свою очередь, используются для обучения мета-классификатора. Последний выполняет объединение решений базовых классификаторов. Однако, в некоторых вариантах этот метод "выходит" за пределы частного случая "stacked generalization". Например, он допускает при мета-обучении использовать не только мета-данные, но также и данные, использованные для обучения базовых классификаторов [36]. Отличие данного алгоритма от других методов, использующих идею "stacked generalization", также и в том, что в нем для формирования мета-данных обязательно используются новые данные, т.е. те, что не были использованы для обучения базовых классификаторов. Этот подход был первоначально разработан для решения таких специфических задач, как обнаружение фальшивых транзакций кредитных карт и обнаружение вторжений в компьютерные сети.

В работе [40] предложен вариант метода "stacked generalization″ для случая бинарной классификации, но для случая, когда каждый из классификаторов базового уровня выдает решение в виде вектора вероятностей принадлежности примера обучающих данных к каждому из классов. В этом случае мета-данные представляются в непрерывной шкале, и для мета-обучения авторы предлагают использовать регрессионную модель, предложенную L.Breiman [8]. Автор экспериментально (путем тестирования на 10 различных наборах данных, взятых с *UCI* репозитория [27] показывает, что такой подход приводит к снижению ошибки классификации по сравнению со случаем использования для мета-обучения бинарных данных, как это имеет место в классической схеме. Однако сравнение ведется с методом простого большинства голосов, который является не самым "хорошим" из методов комбинирования решений классификаторов. Очевидно, что вместо регрессионной модели L.Breiman возможно использование и других подходов к мета-обучению, что может привести к улучшению свойств метода.

Одной из наиболее ранних работ, в которой для комбинирования решений используется идея "stacked generalization", является работа [28]. Большинство авторов, занимающихся проблемой комбинирования решений [28, 7, 18 и др.], подчеркивают, что для обеспечения некоррелированности ошибок, даваемых разными классификаторам базового уровня, целесообразно использовать различные алгоритмы классификации. Например, в работе [28] исходное множество классификаторов строится на базе таких известных и различных алгоритмов, как CN2 [11], C4.5 [37], OC1 [32], PEBLS [14], метод ближайшего соседа, наивный байесовский метод и метод обратного распространения ошибки [36]. Исследования показали, что в этом случае удается добиться лучшего качество классификации по сравнению со случаем, когда используется один и тот же алгоритм, обученный на различных подмножествах данных, как это делается в таких известных методах, как *bagging* [7], *boosting* [17] и в ряде других. Еще одна особенность данной работы состоит в том, что для мета-обучения объединению классификаторов автор использует специальное преобразование этих данных в

[1] Рис.12 как раз соответствует методу мета-классификации.

Труды СПИИРАН. 2002. Вып. 1. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2002. Issue 1. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

непрерывное пространство (в основе такого преобразования лежит сингулярное разложение специальным образом построенной матрицы невязок) с последующим использованием метода ближайшего соседа. Последний шаг необходим для того, чтобы выбрать, какой именно классификатор (здесь — и алгоритм обучения) дает наилучшие результаты в той или иной окрестности обучающих данных. Построенный таким образом метод комбинирования решений известен под аббревиатурой SCANN ("*Stacking, Correspondence Analysis, Nearest Neighbor*"). Таким образом, этот метод по сути комбинирует идею *stacked generalization* и идею оценки компетентности классификаторов, которая лежит в основе методов третьей группы методов комбинирования классификаторов, рассматриваемой далее.

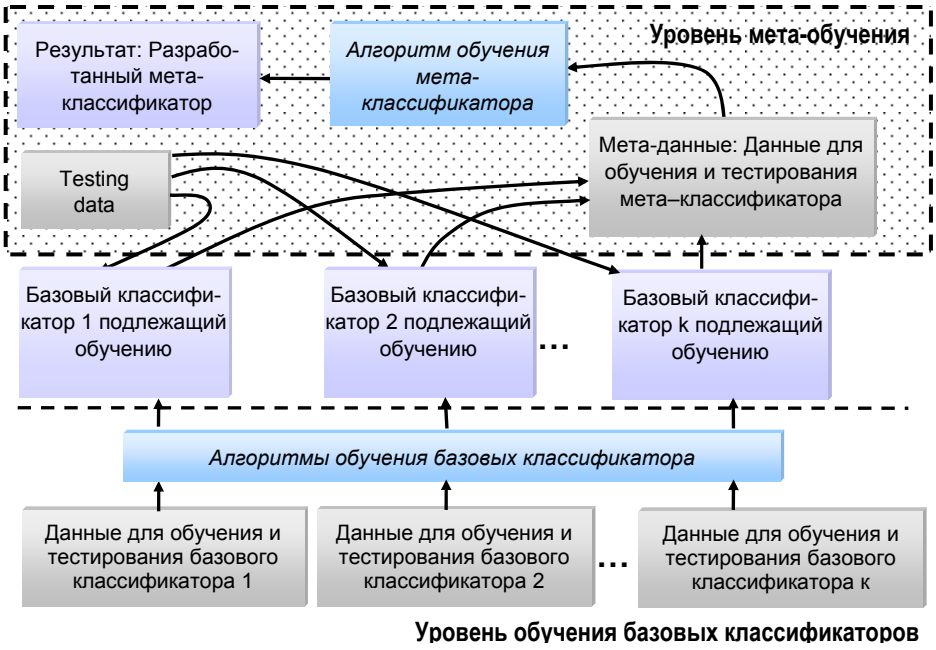

Рис.12. Схема обучения базовых классификаторов, формирования мета-данных и обучения мета-классификатора для комбинирования решений на основе обобщения по типу "матрешки" ("*stacked generalization*")

В работе [18] предлагается еще один метод комбинации решений, который базируется также на идеях "stacked generalization". Он назван методом "каскадного обобщения" ("*cascade generalization*"). Метод содержит ряд оригинальных идей, в частности, он расширяет идеи некоторых из известных методов, которые базируются на идее "stacked generalization".

Идея метода состоит в следующем. В нем несколько различных алгоритмов классификации используются последовательно. Как и в методе, рассмотренном в работе [42], в качестве выхода процедуры "stacked generalization" на базовом уровне рассматривается распределение вероятностей на множестве классов возможных решений, даваемых классификаторами базового уровня В методе "cascade generalization" в составе вектора мета-данных, который строится для каждого из примеров обучающей и тестовой выборок базового уровня, формируются сразу две выборки мета-данных, назовем их ***L*** и ***T***. В них, в отличие от работы [42], сохраняются не только полученные распределения вероятностей, даваемые классификаторами базового уровня, но также и полный набор атрибутов базового уровня. Заметим, что в основной версии алгоритма на базовом уровне используется только один классификатор, хотя в общей модели допускается, что их может быть несколько. Построенные данные–выборки ***L***

Труды СПИИРАН. 2002. Вып. 1. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2002. Issue 1. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

и ***T***, рассматриваются соответственно как обучающая и тестовая выборки для классификаторов мета-уровня (уровня 1), и для них далее решается такая же задача, как и для базового уровня. Одно из основных требований здесь состоит в том, чтобы классификатор мета-уровня базировался на ином алгоритме, чем классификатор базового уровня. Схема каскадного обобщения может включать в себя несколько уровней, причем на каждом из них процесс обучения реализуется аналогичным образом.

В описанных алгоритмах мета-обучения алгоритм мета-классификации может рассматриваться как арбитр, который, рассматривая решения множества классификаторов, может согласиться с кем-либо из них, а может выбрать решение, отличное от всех предложенных. Одним из простых примеров алгоритма мета-классификации, в котором явно используется термин "арбитраж" является алгоритм, рассмотренный в работе [33]. В нем рассматривается задача комбинирования решений при наличии трех классификаторов, один из которых выступает в роли арбитра, когда два других расходятся в своих решениях. Этот же термин используется и в работах по мета-классификации [10, 36].

В целом, группа методов комбинирования решений на основе идеи "stacked generalization*"* является достаточно популярной в среде исследователей, и работы в этом направлении продолжаются. Одним из недостатков методов такого типа считается его неспособность подстраиваться динамически под конкретную задачу, неспособность учитывать конкретные особенности задачи [45].

Этот недостаток пытаются преодолеть в другой группе методов, в которых также иногда используется термин "арбитраж" ("рефери"), однако он используется для других целей. Эти методы базируются на использовании *оценок компетентности классификаторов*. Идея методов этой группы комбинирования решений различных классификаторов состоит в том, чтобы для каждого классификатора базового уровня построить область его "компетентности", т.е. область в пространстве атрибутов, в которой он работает надежнее, чем другие алгоритмы. Определение области компетентности каждого алгоритма выполняется с помощью специальной программы, которая в работе [34] называется "рефери", или "арбитром". Далее задача классификации решается на основе следующего принципа: каждый алгоритм используется там, где он наиболее компетентен, причем только один он. По-видимому, эта идея ранее других была высказана в работах [41] и [28]. Суть метода поясняется на рис. 13.

Многие авторы, которые предлагают алгоритмы комбинирования классификаторов базового уровня на основе оценок их компетентности, опираются на результаты, полученные в работе [6]. Эта работа представляет интерес с различных точек зрения. Основной результат этой работы– это метод для оценивания ошибки конкретной модели классификации, а также модели для оценки различия между моделями различных классификаторов. Как указывают авторы, первая модель может использоваться для таких целей как (1) определение областей в пространстве признаков, где классификатору верить нельзя, (2) определение "трудных" областей данных для конкретного приложения, (3) отыскание областей, в которых классификатор нужно улучшать или "латать", (4) определение влияния изменений в классификаторе на качество его работы.

Очевидно, что при реализации любого метода комбинирования классификаторов базового уровня на основе оценок их компетентности необходимо решать следующие две ключевые проблемы: (1) что понимать под компетентностью алгоритма (автор работы [41] называет это свойство "точностью предска-

Труды СПИИРАН. 2002. Вып. 1. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2002. Issue 1. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

зания в данной области"–"*predictive accuracy in these region*"), и (2) как вычислять этот показатель компетентности для конкретной области и конкретного алгоритма? В работе [41] этот вопрос решается для частных случаев алгоритмов классификации, а именно, для таких известных алгоритмов как наивный байесовский метод и метод ближайшего соседа.

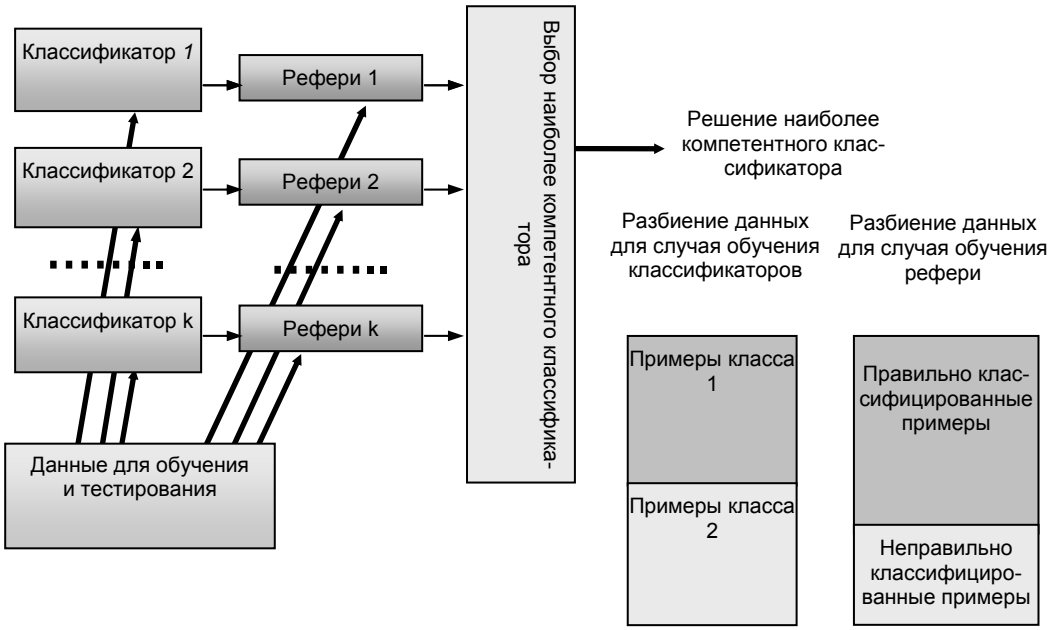

Рис.13. Схема комбинирования решений на основе выбора наиболее компетентного классификатора и схема разбиения данных для обучения классификаторов и для обучения рефери

Более детальное исследование рассматриваемого метода комбинирования решений дано в работе [34]. В ней предполагается, что каждому классификатору ставится в соответствие другой алгоритм, называемый "рефери", который предназначен для оценки надежности (компетентности) соответствующего классификатора. Под надежностью классификатора в заданной области авторы понимают вероятность правильной классификации объектов в ней. Поскольку авторы исследуют этот подход экспериментально применительно к алгоритму C4.5, который строит решающие деревья, то проблема определения компетентности алгоритма решается довольно тривиально — искомая вероятность равна вероятности правильной классификации, поставленной в соответствие тому листу дерева, в который "попадает" объект, подлежащий классификации. Тогда поиск наиболее компетентного классификатора сводится к нахождению того из них, для которого эта вероятность наибольшая.

Общая схема обучения в алгоритме комбинирования классификаторов, предложенном в работе [34], состоит из двух шагов. На первом шаге выполняется обучение и тестирование нескольких классификаторов. Этот шаг ничем не отличается от обычной процедуры обучения. Здесь могут использоваться различные методы, релевантные специфике задачи и исходным данным. Далее по результатам тестирования для каждого классификатора строится разбиение множества обучающих данных на два подмножества L^+ и L^-, где первое подмножество отвечает множеству правильно классифицированных примеров, а второе — множеству ошибочно классифицированных примеров. Рассматривая эти множества данных в качестве областей компетентности и некомпетентности

Труды СПИИРАН. 2002. Вып. 1. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2002. Issue 1. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

классификатора соответственно, их можно использовать в качестве обучающих данных для обучения алгоритма "рефери". Очевидно, что это стандартная задача обучения, которую можно решать различными методами. Нестандартным элементом является только необходимость вычисления вероятности правильной классификации. При классификации новых данных задача рефери состоит в том, чтобы для каждого входного примера определить, принадлежит ли он области компетентности алгоритма или нет, и если принадлежит, то какова вероятность правильной классификации этого примера.

Большинство известных методов классификации позволяют определить вероятность правильной классификации отдельного примера, хотя, возможно, в ряде случаев для этого потребуются некоторые доработки алгоритма. Как уже отмечалось, для метода C4.5 эта проблема решается тривиально [34]. То же самое можно сказать и о "наивном байесовском классификаторе" [41].

Общее свойство методов комбинирования решений на основе оценок компетентности алгоритмов состоит в том, что чем детальнее описаны области компетентности каждого из классификаторов, тем большей точностью обладает алгоритм в целом[1]. В различных методах необходимая степень детальности описания областей компетентности алгоритмов достигается различным образом. В самом простом случае эти характеристики можно получить по результатам тестирования. Хотя такие характеристики компетентности алгоритма могут оказаться достаточно грубыми, но, тем не менее, их использование ведет к повышению точности комбинированного алгоритма по сравнению с точностью, даваемой каждым из них в отдельности [34].

В несколько более поздней работе [39] рассматривается еще одна постановка рассматриваемой задачи, которая во многом сходна с постановкой задачи в работе [34], однако имеет и свои отличия. В частности, в ней задача мета-обучения рассматривается в более общей форме. В этой постановке для каждого классификатора базового уровня после его обучения использованная обучающая выборка разбивается на две подобласти (как и в работе [34]), которые рассматриваются как обучающие данные для алгоритма типа "рефери". Отличительная особенность данного алгоритма состоит в том, что если несколько классификаторов базового уровня оказываются компетентными для одного и того объекта, но дают различные решения, то далее используется либо схема голосования, либо выбор решения делается на основе подсчета меры уверенности, которая вычисляется упрощенным образом по выборке данных на основе частотных оценок вероятностей классов. Эмпирические оценки точности такого подхода показали, что на некоторых данных этот метод превосходит другие методы комбинирования решений.

Другой, весьма интересный, вариант алгоритма данного класса предложен в работах [43] и [44]. В них рассматривается задача комбинирования решений множества классификаторов базового уровня, обученных на одном и том же множестве данных, но построенных с помощью различных алгоритмов обучения. Полагается, что решение каждого такого классификатора, формируемое в процессе тестирования, включает в себя имя класса с некоторой мерой уверенности, например, с вероятностью принадлежности классифицируемого объекта указанному классу. Эти решения, построенные на тестовых данных и записанные в виде множества строк, каждая из которых содержит указанные пары для всех классификаторов базового уровня, образуют мета-данные. Каждой

[1] В пределе, когда достигается достаточно детальное дробление пространства значений атрибутов, метод вырождается в известную схему "*case-based reasoning*".

Труды СПИИРАН. 2002. Вып. 1. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2002. Issue 1. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

строке мета-данных, как обычно, приписывается имя истинного класса. Мета-данные, построенные таким образом, используются для обучения мета-дерева решений. Мета-дерево решений имеет "почти" обычную структуру и строится обычным образом, например, с использованием алгоритма C4.5. Основное отличие мета-дерева от обычного дерева решений состоит в том, что в мета-дереве промежуточным узлам, как обычно, приписываются тестовые атрибуты из числа атрибутов мета-данных, однако в отличие от обычного дерева решений, *каждому листу мета-дерева приписывается имя алгоритма* базового уровня, который классифицирует тестируемый пример с наибольшей мерой уверенности. Напомним, что в "обычном" дереве решений листу приписывается имя класса принадлежности тестового примера. Последнее и является тем новшеством, которое придает данному алгоритму новые свойства. Иначе говоря, мета-дерево задает имя предпочтительного классификатора для подмножества данных, отвечающих листу мета-дерева. Очевидно, что для построения мета-дерева решений можно использовать не только алгоритм C4.5, но и любой другой метод построения дерева решений.

Результаты сравнения различных вариантов построения мета-дерева решений, а также результаты сравнения его свойств с другими методами комбинирования решений, которые получены на основании тщательно проведенного обширного эксперимента, таковы [44]:

1. Для построения мета-дерева предпочтительнее использовать мета-атрибуты, например, вероятности (меры уверенности) правильности принятия решений классификаторами базового уровня, чем атрибуты базового уровня.
2. Алгоритм по своим качествам превосходит алгоритм, использующий решающее дерево [34].
3. Чем более разнообразно множество классификаторов базового уровня, тем лучше работает мета-дерево классификации[1].
4. Предложенный подход превосходит такие широко известные методы, как "*bagging*" и "*boosting*" методы комбинирования решений.

В целом, группа алгоритмов комбинирования решений, использующих рефери для определения наиболее компетентного классификатора в зависимости от входных данных, представляется весьма перспективной. Достоинство такого подхода, кроме повышения точности классификации, также и в том, что введение нового классификатора не потребует переучивания остальных классификаторов. В этом случае ему просто будет назначена область компетентности, в которой он ответственен за принятие решений. Это заключение справедливо как для алгоритмов, описанных в работе [34], так и для алгоритмов, использующих мета-дерево решений [43, 44]. К сожалению, во всех работах оставлен в стороне вопрос о том, как контролировать покрытие всего пространства атрибутов областями компетентности. Косвенным образом влияние этого недостатка снижается требованием некоррелированности ошибок используемых классификаторов.

В заключение следует сказать, что многие из существующих методов комбинирования решений различных классификаторов могут быть эффективно использованы в задаче обучения слиянию данных. Однако, задача слияния данных имеет свои особенности, которые влекут необходимость модификации из-

[1] Этот вывод известен из более ранних работ (см. выше).

Труды СПИИРАН. 2002. Вып. 1. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2002. Issue 1. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

вестных алгоритмов и дальнейших разработок в данном направлении. Один из таких методов рассматривается в следующем подразделе.

3.4. Метод объединения решений частных классификаторов

Дадим описание общей идеи разработанного авторами метода комбинирования решений, ориентированного на решение задач распределенного обучения в задаче слияния данных, который предполагается использовать наряду с некоторыми из ранее описанных. В основе его использованы базовые идеи метода, предложенного в работе [44].

Пусть рассматривается некоторый источник данных. Полагаем, что решения классификаторов базового уровня (они формируют данные для обучения рефери, т.е. алгоритма оценки областей компетентности классификаторов), представлены в формате, приведенном в табл. 1. В ней данные для обучения рефери представлены набором строк, каждая из которых вычислена по некоторому примеру тестовых данных. Каждая строка этих данных состоит из двух компонент. *Первая компонента* — это набор вероятностей правильной классификации (или значений другой меры уверенности) для каждого из используемых базовых классификаторов на соответствующем тестовом примере. Напомним, что данные источников, которые используются при построении данных для обучения рефери, — это выделенный набор данных, в котором строка данных использует все множество атрибутов. *Вторая компонента* этой строки — это список булевых переменных, поставленных в соответствие используемым классификаторам базового уровня. Обозначим алгоритм рефери для алгоритма C_j символом R_j. Для некоторого примера данных для обучения рефери (табл. 1) в столбце, поставленном в соответствие классификатору C_j [1] *j=1,2…K*, будет помещен символ "*1*", если этот классификатор выдал правильное решение для соответствующего тестового примера (для этого примера классификатор C_j "компетентен" принимать решения), и символ "*0*" – в противном случае. В табл. 1 символ $\widetilde{R}_{ij} \in \{0,1\}$, где $\widetilde{R}_{ij}$*=1*, если классификатор R_j выдает правильное решение для примера X_i, и $\widetilde{R}_{ij}$*=0* — в противном случае.

Для обучения рефери могут использоваться различные процедуры. В разрабатываемом программном продукте для этого используется **VAM** ("*Visual Analytical Mining*") алгоритм обучения [22, 2]. В этом алгоритме база знаний рефери строится как база правил, представленных на языке фрагмента исчисления предикатов первого порядка (без кванторов) с линейными термами. При этом каждое из правил, получаемых в результате обучения, имеет в качестве заключение либо R_j с приписанной вероятностной мерой уверенности, если оно аргументирует в пользу компетентности классификатора, или $\overline{R}_j$ (тоже с мерой уверенности), если оно аргументирует в пользу его некомпетентности. Таким образом, набор правил рефери состоит из двух групп, одна из которых "аргументирует" в пользу компетентности классификатора, а другая — в пользу его некомпетентности. Они образуют базу знаний для оценки условных вероятностей $p(\widetilde{R}_{ij} = R_{ij} / X = X_i)$ и $p(\widetilde{R}_{ij} = \overline{R}_{ij} / X = X_i)$ соответственно. Механизм вывода заключения о компетентности классификатора для каждого входного данного строится на основе модели Алгебраической байесовской сети, описанной в

[1] Для простоты записи здесь опущен индекс, указывающий номер источника данных.

Труды СПИИРАН. 2002. Вып. 1. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2002. Issue 1. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

работах [20], [1]. Он основан на вычислении условных вероятностей $p(\widetilde{R}_{ij} = R_{ij} / X = X_i)$, $p(\widetilde{R}_{ij} = \overline{R}_{ij} / X = X_i)$ и последующем расчете апостериорных вероятностей классов.

Таким образом, в результате процедуры обучения каждому классификатору C_j будет поставлен в соответствие алгоритм R_j распознавания его компетентности применительно к каждому конкретному входному данному X_i.

Таблица 1. Формат данных для обучения алгоритма рефери, используемого для объединения решений базовых классификаторов на основе оценки их компетентности

Примеры	$p(C_1)$	$p(C_2)$	....	$p(C_K)$	R_1	R_2	....	R_K
X_1	p_{11}	p_{12}	…	p_{1K}	$\widetilde{R}_{11}$	$\widetilde{R}_{12}$	…	$\widetilde{R}_{1K}$
X_2	p_{21}	p_{22}	…	p_{2K}	$\widetilde{R}_{21}$	$\widetilde{R}_{22}$	…	$\widetilde{R}_{2K}$
…….	…….	…….	…….	…….	…….	…….	…….	…….
X_m	p_{m1}	p_{m2}	…	p_{mK}	$\widetilde{R}_{m1}$	$\widetilde{R}_{m2}$	…	$\widetilde{R}_{mK}$

В отличие от метода оценки компетентности, где используется мнение наиболее компетентного классификатора, предложенного в работах [43, 44], в разработанном методе предполагается объединять решения нескольких наиболее компетентных классификаторов, что позволяет достичь повышения точности итогового решения. Целесообразность такого подхода обусловлена тем, что "компетентные алгоритмы" могут выдавать и противоречивые решения. По этой причине дополнительно возникает задача обучения объединению решений рефери, суть которой в формировании правил совместного использования решений различных компетентных рефери для выработки совместного итогового решения[1]. По своей сути эта задача мало чем отличается от задачи обучения комбинированию решений, а потому может решаться аналогично последней.

Данный алгоритм исследован экспериментально на данных KDD Cup-99 [26] и показал себя перспективным[2].

4. Архитектура многоагентной инструментальной системы для разработки приложений в области слияния данных и архитектура типового агента

Концептуальные аспекты разработки систем слияния данных, описанные в предыдущем материале данной работы, составляют основу разрабатываемой технологии создания приложений в области систем слияния данных. Конечной целью работы является создание программного инструментального средства поддержки этой технологии, которое позволило бы автоматизировать основные операции по созданию приложений. Поскольку любое приложение в данной области по своей сути является распределенным, а участники процесса разра-

[1] В известных алгоритмах комбинирования классификаторов на основе оценки их компетентности в расчет принимается решение только одного "наиболее компетентного" классификатора, определяемого множеством рефери. Именно потому в них не рассматривается процедура обучения совместному объединению решений классификаторов и "мнений" их рефери.

[2] Метод пока еще исследован недостаточно, поэтому его сравнение с другими методами, которые используются в аналогичных целях, пока невозможно.

Труды СПИИРАН. 2002. Вып. 1. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2002. Issue 1. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

ботки также образуют распределенную команду, то естественным решением является выбор многоагентной архитектуры для реализации названного выше программного инструментария.

По существу, основная часть этой архитектуры естественным образом определяется технологией распределенной обработки онтологии и ее компонент, специфических для задач слияния данных, о которых шла речь в разделе 2. В частности, те компоненты, которые представлены на рис. 3 и 5, и схема их взаимодействия, и формируют основную часть архитектуры. В этой части отсутствуют только те компоненты, которые отвечают непосредственно за процесс обучения решателей, а также компоненты целевой системы, которая и является объектом разработки. Схематическое представление архитектуры инструментария для разработки приложений в области слияния данных, а также схема взаимодействия инструментария и результирующей системы в процессе создания и функционирования последней, представлены на рис. 14, 15 и 16. Дадим к ним краткие пояснения.

Многоагентная архитектура инструментальной системы для разработки приложений в области слияния данных состоит из: (1) компонент, которые отвечают за "работу" с отдельными источниками данных, и (2) компонент, ответственных за координацию работы компонент локальных источников, а также за формирование и хранение компонент общей онтологии, базы знаний, и механизмов, ответственных за объединение решений локальных источников. Архитектура компонент, которые отвечают за "работу" с отдельным источником данных, представлена на рис.14. Архитектура компоненты мета-уровня представлена на рис. 15. Оба рисунка представляют также распределение задач обучения объединению данных между отдельными компонентами.

Рассмотрим сначала распределение функций для компоненты архитектуры, ассоциированной с локальным источником (рис. 14).

Агент управления данными локального источника ответственен за разработку своей части локальной онтологии. Он также участвует в разработке общей онтологии приложения. Кроме того, он исполняет роль шлюза, через который осуществляется доступ к данным источника.

*Агент обнаружения знаний имеет целью о*бучение и тестирование агентов локального источника (классификаторов, мета-классификаторов, рефери). Иначе говоря, он решает задачу разработки баз знаний и механизмов принятия решений агентов, работающих с данными локального источника.

Агенты принятия решений (агенты–классификаторы) являются компонентами разрабатываемого приложения, и обучение этих агентов является основной задачей локальных агентов обнаружения знаний (см. выше). Таких агентов в подсистеме, работающей с отдельным источником, может быть несколько, среди них могут быть и агенты мета-классификации и рефери, ответственные за комбинирование решений агентов базового уровня.

Сервер (библиотека) методов обучения включает в себя множество методов извлечения знаний из данных, которые используются по мере необходимости агентами обнаружения знаний.

Функции, которые в разработанной архитектуре возлагаются на компоненту мета-уровня (рис. 15), таковы:

*Агент KDD мастер ответственен за а*нализ структур данных локальных источников, реализацию протокола разработки общей части прикладной онтологии и поддержание ее непротиворечивости, поддержку разработки многоуровневой структуры объединения решений, за управление распределенными

Труды СПИИРАН. 2002. Вып. 1. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2002. Issue 1. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

данными для обучения и тестирования базовых решателей, за формирование мета-данных для обучения решателей мета-уровня.

Агент обнаружения знаний в данных на мета-уровне ответственен за обучение и тестирование агента (или агентов) мета-классификации и/или агента-рефери, т.е. формирование базы знаний и механизма принятия решений на мета-уровне.

Агент мета классификации является компонентой разрабатываемого приложения и его обучение является основной задачей *агента обнаружения знаний в данных на мета-уровне.* Это обучаемый агент мета-уровня.

*Сервер (библиотека) методов обучения с*одержит методы извлечения знаний из данных, метрики, используемые для оценки качества обучения и методы их вычисления.

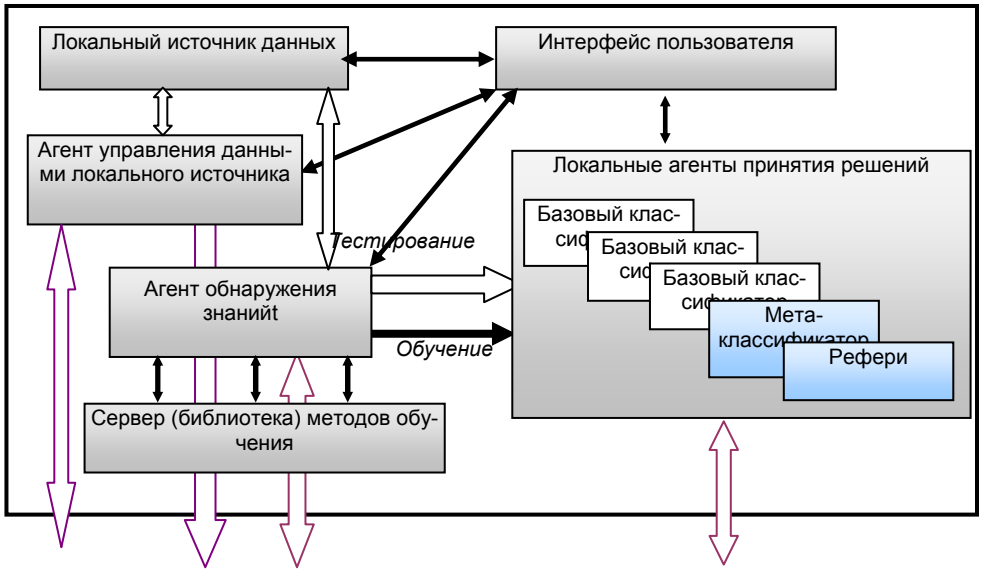

Рис.14. Архитектура многоагентной системы обучения слиянию данных: архитектура подсистемы, которая отвечает за "работу" с локальным источником данных

Труды СПИИРАН. 2002. Вып. 1. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2002. Issue 1. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

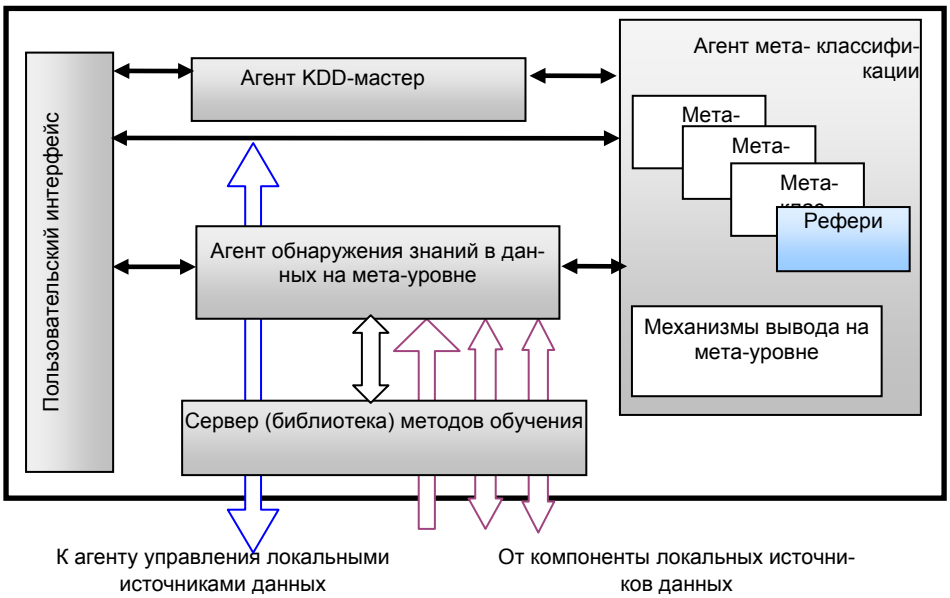

Рис.15. Архитектура многоагентной системы обучения слиянию данных: Подсистема мета-уровня

В соответствии с принятой технологией разработки и программной реализации приложений в области слияния данных, инструментарий рассматривается также как часть приложения, которая используется в процессе всего жизненного цикла системы слияния данных. Схема их взаимодействия представлена на рис.16. На начальном этапе инструментарий используется для разработки приложения. На этапе эксплуатации приложения инструментарий используется для модификации приложения, в частности для модификации баз знаний классификаторов и механизмов классификации с целью использования полученного опыта для улучшения качества работы системы.

Для всех агентов, присутствующих в архитектуре инструментария, используется одна и та же типовая архитектура, которая представлена на рис.17. Различия агентов, отражающие их специализацию в общей архитектуре инструментальной системы, представляются содержанием их баз данных и знаний, в частности, алгоритмами, реализуемыми машинами состояний.

Окружение агента составляют: (1) другие агенты, с которыми он обменивается сообщениями, (2) внешняя среда, параметры состояния которой воспринимаются, и возможно, изменяются агентом, а также (3) пользователь, с которым агент общается через пользовательский интерфейс. Прием и посылка сообщений выполняются компонентами *Протокол приема сообщений* и *Протокол отправки сообщений* соответственно. Полученные сообщения записываются в *Буфер входящих сообщений*. Порядок их выбора из буфера для обработки определяется компонентой *Обработчик входных сообщений*. Она же исполняет роль синтаксического анализатора, который интерпретирует KQML сообщение и извлекает из него содержание. Компонента *База данных диалогов агента* ведет учет поступивших и отосланных сообщений и хранит некоторые их атрибуты (идентификатор сообщения, тип сообщения), а также сопоставляет каждому входному (выходному) сообщению, требующему ответа, соответствующее выходное (входное) сообщение, когда оно будет сформировано и отослано (получено). *Мета-машина состояний* управляет процессами семантической обработки входных сообщений, передавая их на обработку соответствующим *Ма-*

Труды СПИИРАН. 2002. Вып. 1. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2002. Issue 1. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

шинам состояний. В процессе управления *Мета-машина состояний* определяет порядок функционирования *Машин состояний* и выделяет на это соответствующие ресурсы процессора в соответствии с приоритетами решаемых задач. В зависимости от истории работы агента изменяется его внутреннее состояние. Компонента *Машина состояний самоактивации агента* при передаче ей управления со стороны *Мета-машины состояний* осуществляет анализ внутреннего состояния агента, и в результате этого может вызвать запуск функциональностей агента, которые не связаны непосредственно с обработкой какого-то конкретного входящего сообщения или с изменением состояния внешней среды. Кроме того, выполнение функциональностей агента может быть инициировано конечным пользователем агента. При этом используется компонента *Пользовательский интерфейс.*

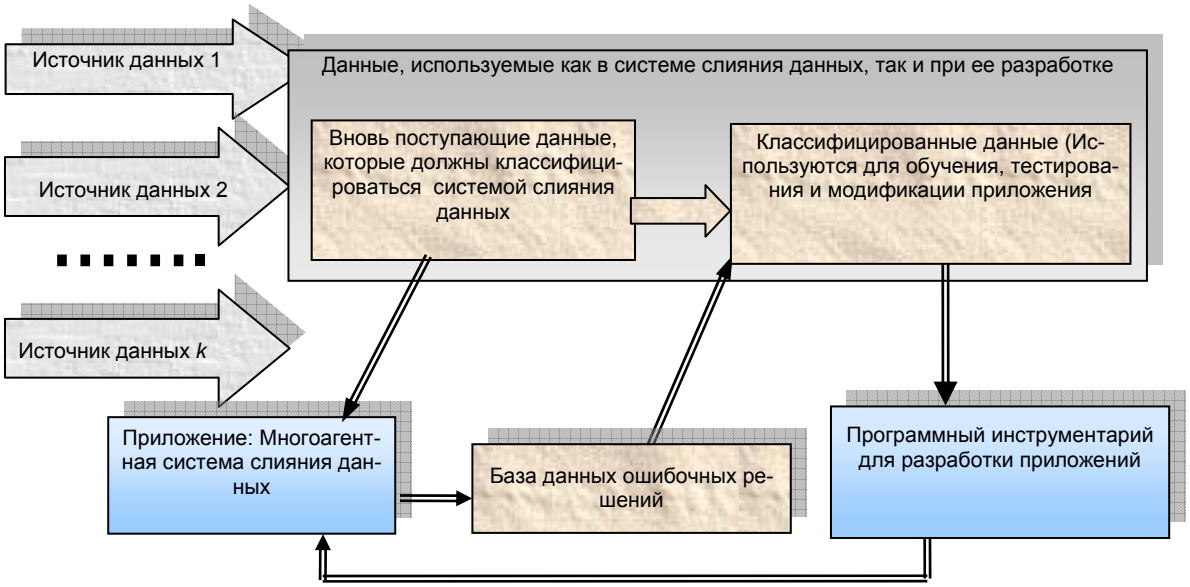

Рис.16. Взаимодействие приложения и инструментальной системы

5. Заключение

В данной работе проанализированы технологические особенности задач принятия решений на основе объединения данных, полученных из различных источников. Эти задачи называются кратко *задачами объединения ("слияния") данных*.

Основные проблемы, которые необходимо решать при проектировании и разработке систем, решающих задачи названного класса, обусловлены распределенностью источников данных и гетерогенностью" данных, представленных в источниках, что влечет необходимость включения в технологию обработки данных ряда новых процедур (по сравнению с технологией разработки других систем поддержки принятия решений), а в программную реализацию — новых компонент. В частности, в рамках задач рассматриваемого класса существенно повышается роль онтологии, однако при этом существенно повышается также и сложность ее создания. В данном случае проблема создания общей онтологии приложения осложняется тем фактом, что в ее разработке должны участвовать распределенные пользователи, что затрудняет процесс поддержания непротиворечивости ее компонент.

Труды СПИИРАН. 2002. Вып. 1. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2002. Issue 1. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

В работе представлена мета-модель источников данных и технология ее разработки. Эта модель обеспечивает единый "взгляд" на данные с мета-уровня, хотя в источниках данные могут иметь иное представление. Разработана также мета-модель объединения решений распределенных классификаторов, структурированных в несколько уровней. Данные модели и технология их создания формируют основу технологии систем слияния данных.

Одной из целей данной работы является создание программной инструментальной системы поддержки технологии разработки приложений в области слияния данных. По нашему мнению адекватной архитектурой для такой инструментальной системы может быть архитектура, использующая парадигму многоагентных систем. Такая архитектура предложена в данной работе. В ней предполагается, что она, помимо традиционных функций обучения, поддерживает также реализацию функций формирования онтологии для конкретного приложения, а также ряд специальных функций работы с данными распределенных источников.

В настоящий момент основные компоненты представленной многоагентной инструментальной системы разработаны и реализованы программно. С ее использованием разрабатывается многоагентная система обучения обнаружению вторжений в компьютерную сеть, а также ряд других приложений.

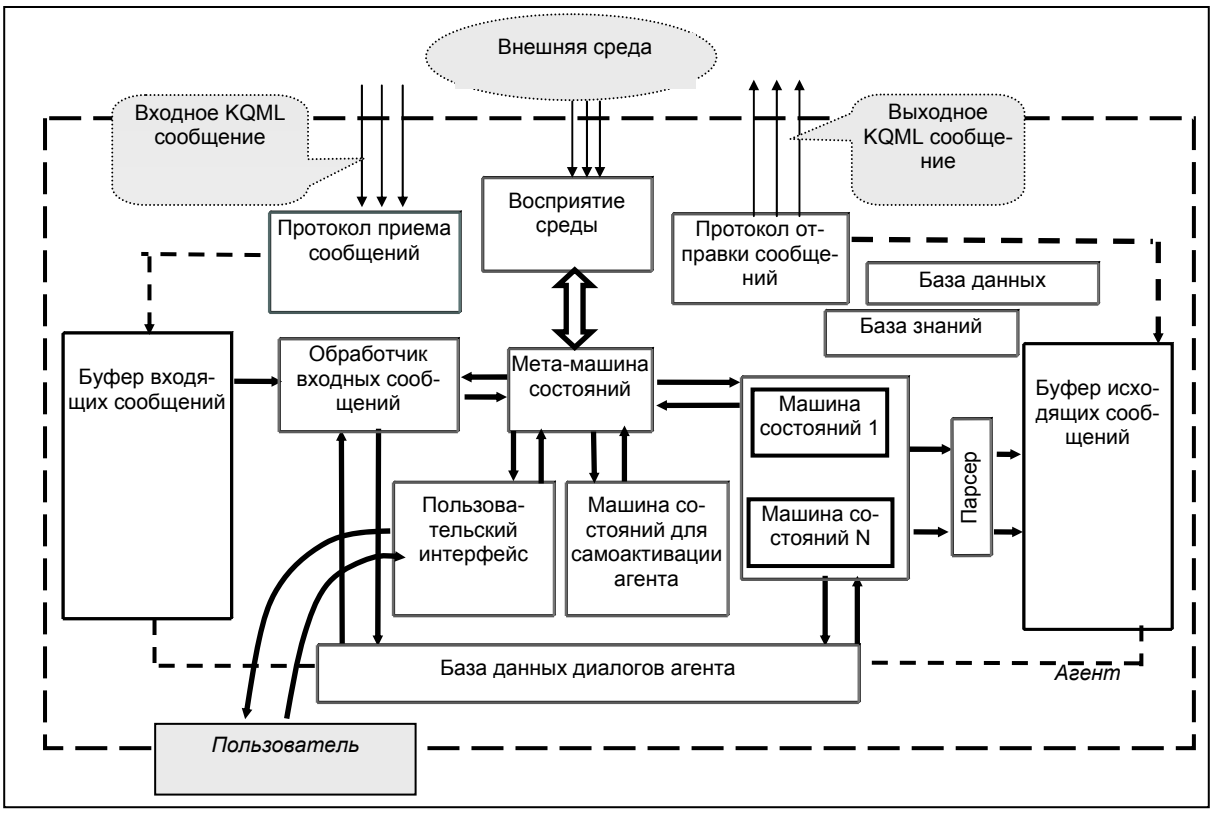

Рис.17. Типовая архитектура агента

Литература

[1] *Городецкий В., Тулупьев А.*. Непротиворечивость баз знаний с интервальной вероятностной мерой неопределенности. Известия РАН "Теория и системы управления", №5, 1997, — с. 23-32.

Труды СПИИРАН. 2002. Вып. 1. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2002. Issue 1. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

[2] *Городецкий В., Самойлов. В.* Визуальный синтез классифицирующих предикатов и их применение для мета-классификации // Труды Таганрогского радиотехнического института, 4, 2001, — с. 5–16.

[3] *Растригин Л., Эренштейн Р.* Методы коллективного распознавания. Москва, Энергоиздат, 1981, — 102 с.

[4] *Ali K. and Pazzani M.* Error reduction through learning multiple descriptions // Machine Learning, 24(3), 1996. — с. 173–202

[5] *Bass T.* Intrusion Detection System and Multisensor Data Fusion: Creating Cyberspace Situational Awareness // Communication of the ACM, vol. 43, no. 4, 2000. — p. 99–105.

[6] *Bay S. and Pazzani M.* Characterizing model error and differences // Proceedings of 17th International Conference on machine learning (ICML–2000), Morgan Kaufmann, 2000.

[7] *Breiman L.* Bagging predictors // Machine Learning, 24 (2), 1996. — p. 123–140.

[8] *Breiman L*. Stacked regression // Machine Learning, 24(1), 1996, — p. 49–64,

[9] *Buntine W.L.* A theory of learning classification rules // Ph.D thesis, University of Technology, School of Computing Science, Sydney, Australia, 1990.

[10] *Chan P.and Stolfo S*. A comparative evaluation of voting and meta-learning on partitioned data // Proceedings of Twelfth 4th International Conference on machine Learning, Tahoe City, CA,1995, — p. 90–98.

[11] *Clark P.and Niblett P.* The CN2 induction algorithm // Machine learning, 3(4), 1989, — p. 261–283.

[12] *Clemen R.* Combining forecasts: A review and annotated bibliography // International Journal of Forecasting, 5, 1989. — p. 559–583.

[13] Corkill D and Lesser V. The use of meta-level control for coordination in distributed problem solving network // Proceedings of the Eighth International Joint Conference on Artificial Intelligence (IJCAI–83), Menlo Park, CA, 1983. —p. 767–770.

[14] *Cost S. and Salzberg S.* A weighted nearest neighbor algorithm for learning with symbolic features // Machine Learning, 10(1), 1993 — p. 57–78.

[15] *Dietterich T.* Machine Learning Research: Four Current Directions // AI magazine. 18(4), 1997. —p. 97–136.

[16] *Dietterich T.* Ensemble Methods in Machine Learning // M.Arbib (Ed.) Handbook of Brain Theory and Neural Networks, 2nd Edition, MIT Press, 2001.

[17] *Freund Y.and Shapire R*. Experiments with a new boosting algorithm // L.Saitta (Ed.) Machine Learning. Proceedings of the 13th International Conference. Morgan Kaufmann, 1996.

[18] *Gama J.and Brazdil P.* Cascade generalization // Machine Learning, 41(3), 2000. — p. 315–342.

[19] *Goodman I., Mahler R., and Nguen H.* Mathematics of Data Fusion // Kluwer Academic Publishers, 1997.

[20] *Gorodetski V.* Bayes' Inference and Decision Making in Artificial Intelligence Systems" // Industrial Applications of Artificial Intelligence. North-Holland, 1991. — p. 276–281.

[21] *Gorodetski V.and Karsayev O.* Algorithm of Rule Extraction from Learning Data // Proceedings of the 8th International Conference "Expert Systems & Artificial Intelligence" (EXPERSYS–96),1996. — p. 133–138.

[22] *Gorodetski V., Skormin V., Popyack L., and Karsaev O.* Distributed Learning in a Data Fusion System // Proceedings of the Conference of the World Computer Congress (WCC–2000) "Intelligent Information Processing" (IIP2000), Beijing, 2000. — p. 147–154.

[23] *Hashem S.* Optimal linear combination of neural networks // Ph.D. thesis, Purdue University, School of Industrial Engineering, Lafaette, IN, 1997.

[24] *Jordan M.and Jacobs R.* Hierarchical mixtures of experts and the EM algorithm // Neural Computations, 6(2), 1994. — p. 181–214.

[25] "KAON – The KArlsruhe Ontology and Semantic Web Infrastructure". `http://kaon.semanticweb.org/`

[26] `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`

[27] *Mason C. and Johnson K.* DATMS: A framework for distributed assumption-based reasoning // Distributed Artificial Intelligence, vol. 2, Eds.M.Hurbis and L.Gasser. CA, Morgan Kaufmann, 1989. — p. 293–318.

[28] *Merz C*. Combining classifiers using correspondence analysis // Advances in Neural Information Processing, 1997.

[29] *Merz C.and Murphy P.* UCI repository of machine learning databases. Irvine, CA, University of California, Department of Information and Computer Science, 1997 `http://www.ics.uci.edu/!mlearn/MLR repository.html.`

[30] *Michalski R.* A Theory and Methodology of Inductive Learning // Machine Learning, vol.1, Eds. J.G.Carbonel, R.S.Michalski, and T.M.Mitchel, Tigoda, Palo Alto, 1983. — p. 83–134.

[31] *Michalski R.and Kaufmann K.* The AQ19 System for Machine Learning and Pattern discovery: A General Description and User's guide. George Mason University. Technical Report ML01–2, P01–2, 2001.

[32] *Murthy S., Kassif S., Salzberg S., and Beigel R.* OC1: Randomized Induction of oblique decision trees // Proceedings of AAAI–93, AAAI Press, 1993.

[33] *Niyogi P., Pierrot J–B., and Siohan O. "*Multiple Classifiers by constrained minimization" // Proceedings of International Conference on Acoustics, Speech, and Signal Processing, Istanbul, Turkey, June, 2000.

[34] *Ortega J., Coppel M., Argamon S.* Arbitraining Among Competing Classifiers Using Learned Referees // Knowledge and Information Systems 3 (2001) 4, 2001. — p. 470–490.

[35] *Perrone M. and Cooper L.* When networks disagree: Ensemble methods for hybrid neural networks // R.J.Mammone (Ed.), Neural Networks for Speech and Image Processing, Chapman and Hall, 1993.

[36] *Prodomidis A., Chan P., and Stolfo S.* Meta–learning in distributed data mining systems: Issues and approaches // P.Chan and H.Kargupta (Eds.) Advances in Distributed Data Mining, AAAI Press, 1999. Also available at `http://www.cs.columbia.edu/~sal/hpapers/DDMBOOK.ps.gz.`

[37] *Quinlan R.* C4.5 Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA, 1993.

[38] *Rumelhart D., Hinton G., and Williams R.* Learning internal representation by error propagation // D.Rumelhart, J.McClelland (Eds.) Parallel Distributed Processing: Exploration of the microstructure of cognitions, Volume 1: Foundations. MIT Press, 1986.

[39] *Seewald A.and Fuernkranz J.* An evaluation of grading classifiers // Proceedings of 4th International Conference "Intelligent data Analysis", LNCS 2189, 2001. — p. 115–124.

[40] Sementic Web roadmap. `http://www.w3.org/DesignIssues/Semantic.html`

[41] *Ting K.* The characterization of predictive accuracy and decision combination // Proceedings of 13th International Conference on Machine Learning, Morgan Kaufman, 1996. — p. 498–506.

[42] *Ting K.and Witten I.* Issues in stacked generalization // Journal of Artificial Intelligence Research, 10, 1999. — p. 271–289.

[43] *Todorovski L.and Dzeroski S.* Combining classifiers with meta decision trees // D.A.Zighen, J.Komarowski and J.Zitkov (Eds.) Proceedings of 4th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD–2000), Springer Verlag, 2000. — p. 54–64.

[44] *Todorovski L.and Dzeroski S.* Combining multiple models classifiers with meta decision trees // To appear in Machine Learning Journal. 2001.

[45] *Vilalta R.and Drissi Y.* A perspective view and survey of meta-learning // Submitted to the Journal of Artificial Intelligence Review, `http://www.research.ibm.com/people/v/vilalta/papers/jaireview01.ps`

[46] *D.Wolpert.* Stacked generalization // Neural Network, 5(2), 1992. — p. 241–260.

Труды СПИИРАН. 2002. Вып. 1. Т. 2. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2002. Issue 1. V. 2. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru