Введение в экспертные системы

Экспертные системы, базовые понятия

Об экспертных системах (ЭС) можно говорить много и сложно. Но наш разговор очень упростится, если мы будем исходить из следующего определения экспертной системы. Экспертная система — это программа (на современном уровне развития человечества), которая заменяет эксперта в той или иной области.

Отсюда вытекает простой вывод — все, что мы изучаем в курсе "Основы проектирования систем с ИИ", конечной целью ставит разработку ЭС. В этой лекции мы остановимся только на некоторых особенностях их построения, которые не затрагиваются в остальных лекциях.

ЭС предназначены, главным образом, для решения практических задач, возникающих в слабо структурированной и трудно формализуемой предметной области. ЭС были первыми системами, которые привлекли внимание потенциальных потребителей продукции искусственного интеллекта.

С ЭС связаны некоторые распространенные заблуждения.

Заблуждение первое: ЭС будут делать не более (а скорее даже менее) того, чем может эксперт, создавший данную систему. Для опровержения данного постулата можно построить самообучающуюся ЭС в области, в которой вообще нет экспертов, либо объединить в одной ЭС знания нескольких экспертов, и получить в результате систему, которая может то, чего ни один из ее создателей не может.

Заблуждение второе: ЭС никогда не заменит человека-эксперта. Уже заменяет, иначе зачем бы их создавали?

Экспертные системы, методика построения

В настоящее время сложилась определенная технология разработки ЭС, которая включает следующие шесть этапов: идентификация, концептуализация, формализация, выполнение, тестирование и опытная эксплуатация.

Рис. 9.1. Методика (этапы) разработки ЭС

Этап идентификации

Этап идентификации связан, прежде всего, с осмыслением тех задач, которые предстоит решить будущей ЭС, и формированием требований к ней. Результатом данного этапа является ответ на вопрос, что надо сделать и какие ресурсы необходимо задействовать (идентификация задачи, определение участников процесса проектирования и их роли, выявление ресурсов и целей).

Обычно в разработке ЭС участвуют не менее трех-четырех человек — один эксперт, один или два инженера по знаниям и один программист, привлекаемый для модификации и согласования инструментальных средств. Также к процессу разработки ЭС могут по мере необходимости привлекаться и другие участники. Например, инженер по знаниям может пригласить других экспертов, чтобы убедиться в правильности своего понимания основного эксперта, представительности тестов, демонстрирующих особенности рассматриваемой задачи, совпадения взглядов различных экспертов на качество предлагаемых решений. Кроме того, для сложных систем считается целесообразным привлекать к основному циклу разработки несколько экспертов. Однако в этом случае, как правило, требуется, чтобы один из экспертов отвечал за непротиворечивость знаний, сообщаемых коллективом экспертов.

Идентификация задачи заключается в составлении неформального (вербального) описания, в котором указываются: общие характеристики задачи; подзадачи, выделяемые внутри данной задачи; ключевые понятия (объекты), их входные (выходные) данные; предположительный вид решения, а также знания, относящиеся к решаемой задаче.

В процессе идентификации задачи инженер по знаниям и эксперт работают в тесном контакте. Начальное неформальное описание задачи экспертом используется инженером по знаниям для уточнения терминов и ключевых понятий. Эксперт корректирует описание задачи, объясняет, как решать ее и какие рассуждения лежат в основе того или иного решения. После нескольких циклов, уточняющих описание, эксперт и инженер по знаниям получают окончательное неформальное описание задачи.

При проектировании ЭС типичными ресурсами являются источники знаний, время разработки, вычислительные средства и объем финансирования. Для эксперта источниками знаний служат его предшествующий опыт по решению задачи, книги, известные примеры решения задач, а для инженера по знаниям — опыт в решении аналогичных задач, методы представления знаний и манипулирования ими, программные инструментальные средства. При определении времени разработки обычно имеется в виду, что сроки разработки и внедрения ЭС составляют, как правило, не менее года (при трудоемкости 5 чел.-лет). Определение объема финансирования оказывает существенное влияние на процесс разработки, так как, например, при недостаточном финансировании предпочтение может быть отдано не разработке оригинальной новой системы, а адаптации существующей.

При идентификации целей важно отличать цели, ради которых создается ЭС, от задач, которые она должна решать. Примерами возможных целей являются: формализация неформальных знаний экспертов; улучшение качества решений, принимаемых экспертом; автоматизация рутинных аспектов работы эксперта (пользователя); тиражирование знаний эксперта.

Этап концептуализации

На данном этапе проводится содержательный анализ проблемной области, выявляются используемые понятия и их взаимосвязи, определяются методы решения задач. Этот этап завершается созданием модели предметной области (ПО), включающей основные концепты и отношения. На этапе концептуализации определяются следующие особенности задачи: типы доступных данных; исходные и выводимые данные, подзадачи общей задачи; применяемые стратегии и гипотезы; виды взаимосвязей между объектами ПО, типы используемых отношений (иерархия, причина — следствие, часть — целое и т.п.); процессы, применяемые в ходе решения; состав знаний, используемых при решении задачи; типы ограничений, накладываемых на процессы, которые применены в ходе решения; состав знаний, используемых для обоснования решений.

Существует два подхода к процессу построения модели предметной области, которая является целью разработчиков ЭС на этапе концептуализации. Признаковый или атрибутивный подход предполагает наличие полученной от экспертов информации в виде троек объект—атрибут—значение атрибута, а также наличие обучающей информации. Этот подход развивается в рамках направления, получившего название "формирование знаний" или "машинное обучение" (machine learning).

Второй подход, называемый структурным (или когнитивным), осуществляется путем выделения элементов предметной области, их взаимосвязей и семантических отношений.

Для атрибутивного подхода характерно наличие наиболее полной информации о предметной области: об объектах, их атрибутах и о значениях атрибутов. Кроме того, существенным моментом является использование дополнительной обучающей информации, которая задается группированием объектов в классы по тому или иному содержательному критерию. Тройки объект—атрибут—значение атрибута могут быть получены с помощью так называемого метода реклассификации, который основан на предположении что задача является объектно-ориентированной и объекты задачи хорошо известны эксперту. Идея метода состоит в том, что конструируются правила (комбинации значений атрибутов), позволяющие отличить один объект от другого. Обучающая информация может быть задана на основании прецедентов правильных экспертных заключений, например, с помощью метода извлечения знаний, получившего название "анализ протоколов мыслей вслух".

При наличии обучающей информации для формирования модели предметной области на этапе концептуализации можно использовать весь арсенал методов, развиваемых в рамках задачи распознавания образов. Таким образом, несмотря на то, что здесь атрибутивному подходу не уделено много места, он является одним из потребителей всего того, что было указано в лекции, посвященной распознаванию образов и автоматического группирования данных.

Структурный подход к построению модели предметной области предполагает выделение следующих когнитивных элементов знаний: 1. Понятия. 2. Взаимосвязи. 3. Метапонятия. 4. Семантические отношения.

Выделяемые понятия предметной области должны образовывать систему, под которой понимается совокупность понятий, обладающая следующими свойствами: уникальностью (отсутствием избыточности); полнотой (достаточно полным описанием различных процессов, фактов, явлений и т.д. предметной области); достоверностью (валидностью — соответствием выделенных единиц смысловой информации их реальным наименованиям) и непротиворечивостью (отсутствием омонимии).

При построении системы понятий с помощью "метода локального представления" эксперта просят разбить задачу на подзадачи для перечисления целевых состояний и описания общих категорий цели. Далее для каждого разбиения (локального представления) эксперт формулирует информационные факты и дает им четкое наименование (название). Считается, что для успешного решения задачи построения модели предметной области число таких информационных фактов в каждом локальном представлении, которыми человек способен одновременно манипулировать, должно быть примерно равно семи.

"Метод вычисления коэффициента использования" основан на следующей гипотезе. Элемент данных (или информационный факт) может являться понятием, если он:

* используется в большом числе подзадач;
* используется с большим числом других элементов данных;
* редко используется совместно с другими элементами данных по сравнению с общим числом его применения во всех подзадачах (это и есть коэффициент использования).

Полученные значения могут служить критерием для классификации всех элементов данных и, таким образом, для формирования системы понятий.

"Метод формирования перечня понятий" заключается в том, что экспертам (желательно, чтобы их было больше двух) дается [задание](https://moodler2.bgpu.ru/mod/assign/view.php?id=109840) составить список понятий, относящихся к исследуемой предметной области. Понятия, выделенные всеми экспертами, включаются в систему понятий, остальные подлежат обсуждению.

"Ролевой метод" состоит в том, что эксперту дается [задание](https://moodler2.bgpu.ru/mod/assign/view.php?id=109840) обучить инженера по знаниям решению некоторых задач предметной области. Таким образом, эксперт играет роль учителя, а инженер по знаниям — роль ученика. Процесс обучения записывается на магнитофон. Затем третий участник прослушивает магнитофонную ленту и выписывает на бумаге все понятия, употребленные учителем или учеником.

При использовании метода "составления списка элементарных действий" эксперту дается [задание](https://moodler2.bgpu.ru/mod/assign/view.php?id=109840) составить такой список при решении задачи в произвольном порядке.

В методе "составление оглавления учебника" эксперту предлагается представить ситуацию, в которой его попросили написать учебник. Необходимо составить на бумаге перечень предполагаемых глав, разделов, параграфов, пунктов и подпунктов книги.

"Текстологический метод" формирования системы понятий заключается в том, что эксперту дается [задание](https://moodler2.bgpu.ru/mod/assign/view.php?id=109840) выписать из руководств (книг по специальности) некоторые элементы, представляющие собой единицы смысловой информации.

Группа методов установления взаимосвязей предполагает установление семантической близости между отдельными понятиями. В основе установления взаимосвязей лежит психологический эффект "свободных ассоциаций", а также фундаментальная категория близости объектов или концептов.

Эффект свободных ассоциаций заключается в следующем. Испытуемого просят отвечать на заданное слово первым пришедшим на ум словом. Как правило, реакция большинства испытуемых (если слова не были слишком необычными) оказывается одинаковой. Количество переходов в цепочке может служить мерой "смыслового расстояния" между двумя понятиями. Многочисленные опыты подтверждают гипотезу, что для двух любых слов (понятий) существует ассоциативная цепочка, состоящая не более чем из семи слов.

"Метод свободных ассоциаций" основан на психологическом эффекте, описанном выше. Эксперту предъявляется понятие с просьбой назвать как можно быстрее первое пришедшее на ум понятие из сформированной ранее системы понятий. Далее производится анализ полученной информации.

В методе "сортировка карточек" исходным материалом служат выписанные на карточки понятия. Применяются два варианта метода. В первом эксперту задаются некоторые глобальные критерии предметной области, которыми он должен руководствоваться при раскладывании карточек на группы. Во втором случае, когда сформулировать глобальные критерии невозможно, эксперту дается [задание](https://moodler2.bgpu.ru/mod/assign/view.php?id=109840) разложить карточки на группы в соответствии с интуитивным пониманием семантической близости предъявляемых понятий.

"Метод обнаружения регулярностей" основан на гипотезе о том, что элементы цепочки понятия, которые человек вспоминает с определенной регулярностью, имеют тесную ассоциативную взаимосвязь. Для эксперимента произвольным образом отбирается 20 понятий. Эксперту предъявляется одно из числа отобранных. Процедура повторяется до 20 раз, причем каждый раз начальные концепты должны быть разными. Затем инженер по знаниям анализирует полученные цепочки с целью нахождения постоянно повторяющихся понятий (регулярностей). Внутри выделенных таким образом группировок устанавливаются ассоциативные взаимосвязи.

Кроме рассмотренных выше неформальных методов для установления взаимосвязей между отдельными понятиями применяются также формальные методы. Сюда в первую очередь относятся методы семантического дифференциала и репертуарных решеток.

Выделенные понятия предметной области и установленные между ними взаимосвязи служат основанием для дальнейшего построения системы метапонятий — осмысленных в контексте изучаемой предметной области системы группировок понятий. Для определения этих группировок применяют как неформальные, так и формальные методы.

Интерпретация, как правило, легче дается эксперту, если группировки получены неформальными методами. В этом случае выделенные классы более понятны эксперту. Причем в некоторых предметных областях совсем не обязательно устанавливать взаимосвязи между понятиями, так как метапонятия, образно говоря, "лежат на поверхности".

Последним этапом построения модели предметной области при концептуальном анализе является установление семантических отношений между выделенными понятиями и метапонятиями. Установить семантические отношения — это значит определить специфику взаимосвязи, полученной в результате применения тех или иных методов. Для этого необходимо каждую зафиксированную взаимосвязь осмыслить и отнести ее к тому или иному типу отношений.

Существует около 200 базовых отношений, например, "часть — целое", "род — вид", "причина — следствие", пространственные, временные и другие отношения. Для каждой предметной области помимо общих базовых отношений могут существовать и уникальные отношения.

"Прямой метод" установления семантических отношений основан на непосредственном осмыслении каждой взаимосвязи. В том случае, когда эксперт затрудняется дать интерпретацию выделенной взаимосвязи, ему предлагается следующая процедура. Формируются тройки: понятие 1 — связь — понятие 2. Рядом с каждой тройкой записывается короткое предложение или фраза, построенное так, чтобы понятие 1 и понятие 2 входили в это предложение. В качестве связок используются только содержательные отношения и не применяются неопределенные связки типа "похож на" или "связан с".

Для "косвенного метода" не обязательно иметь взаимосвязи, достаточно лишь наличие системы понятий. Формулируется некоторый критерий, для которого из системы понятий выбирается определенная совокупность концептов. Эта совокупность предъявляется эксперту с просьбой дать вербальное описание сформулированного критерия. Концепты предъявляются эксперту все сразу (желательно на карточках). В случае затруднений эксперта прибегают к разбиению отобранных концептов на группы с помощью более мелких критериев. Исходное количество концептов может быть произвольным, но после разбиения на группы в каждой из таких групп должно быть не более десяти концептов. После того как составлены описания по всем группам, эксперту предлагают объединить эти описания в одно.

Следующий шаг в косвенном методе установления семантических отношений — это анализ текста, составленного экспертом. Концепты заменяют цифрами (это может быть исходная нумерация), а связки оставляют. Тем самым строится некоторый граф, вершинами которого служат концепты, а дугами — связки (например, "ввиду", "приводит к", "выражаясь с одной стороны", "обусловливая", "сочетаясь", "определяет", "вплоть до" и т.д.) Этот метод позволяет устанавливать не только базовые отношения, но и отношения, специфические для конкретной предметной области.

Рассмотренные выше методы формирования системы понятий и метапонятий, установления взаимосвязей и семантических отношений в разных сочетаниях применяются на этапе концептуализации при построении модели предметной области.

Этап формализации

Теперь все ключевые понятия и отношения выражаются на некотором формальном языке, который либо выбирается из числа уже существующих, либо создается заново. Другими словами, на данном этапе определяются состав средств и способы представления декларативных и процедурных знаний, осуществляется это представление и в итоге формируется описание решения задачи ЭС на предложенном (инженером по знаниям) формальном языке.

Выходом этапа формализации является описание того, как рассматриваемая задача может быть представлена в выбранном или разработанном формализме. Сюда относится указание способов представления знаний (фреймы, сценарии, семантические сети и т.д.) и определение способов манипулирования этими знаниями (логический вывод, аналитическая модель, статистическая модель и др.) и интерпретации знаний.

Этап выполнения

Цель этого этапа — создание одного или нескольких прототипов ЭС, решающих требуемые задачи. Затем на данном этапе по результатам тестирования и опытной эксплуатации создается конечный продукт, пригодный для промышленного использования. Разработка прототипа состоит в программировании его компонентов или выборе их из известных инструментальных средств и наполнении базы знаний.

Главное в создании прототипа заключается в том, чтобы этот прототип обеспечил проверку адекватности идей, методов и способов представления знаний решаемым задачам. Создание первого прототипа должно подтвердить, что выбранные методы решений и способы представления пригодны для успешного решения, по крайней мере, ряда задач из актуальной предметной области, а также продемонстрировать тенденцию к получению высококачественных и эффективных решений для всех задач предметной области по мере увеличения объема знаний.

После разработки первого прототипа ЭС-1 круг предлагаемых для решения задач расширяется, и собираются пожелания и замечания, которые должны быть учтены в очередной версии системы ЭС-2. Осуществляется развитие ЭС-1 путем добавления "дружественного" интерфейса, средств для исследования базы знаний и цепочек выводов, генерируемых системой, а также средств для сбора замечаний пользователей и средств хранения библиотеки задач, решенных системой.

Выполнение экспериментов с расширенной версией ЭС-1, анализ пожеланий и замечаний служат отправной точкой для создания второго прототипа ЭС-2. Процесс разработки ЭС-2 — итеративный. Он может продолжаться от нескольких месяцев до нескольких лет в зависимости от сложности предметной области, гибкости выбранного представления знаний и степени соответствия управляющего механизма решаемым задачам (возможно, потребуется разработка ЭС-3 и т.д.). При разработке ЭС-2, кроме перечисленных задач, решаются следующие:

* анализ функционирования системы при значительном расширении базы знаний;
* исследование возможностей системы в решении более широкого круга задач и принятие мер для обеспечения таких возможностей;
* анализ мнений пользователей о функционировании ЭС;
* разработка системы ввода-вывода, осуществляющей анализ или синтез предложений ограниченного естественного языка, позволяющей взаимодействовать с ЭС-2 в форме, близкой к форме стандартных учебников для данной области.

Если ЭС-2 успешно прошла этап тестирования, то она может классифицироваться как промышленная экспертная система.

Этап тестирования

В ходе данного этапа производится оценка выбранного способа представления знаний в ЭС в целом. Для этого инженер по знаниям подбирает примеры, обеспечивающие проверку всех возможностей разработанной ЭС.

Различают следующие источники неудач в работе системы: тестовые примеры, ввод-вывод, правила вывода, управляющие стратегии.

Показательные тестовые примеры являются наиболее очевидной причиной неудачной работы ЭС. В худшем случае тестовые примеры могут оказаться вообще вне предметной области, на которую рассчитана ЭС, однако чаще множество тестовых примеров оказывается слишком однородным и не охватывает всю предметную область. Поэтому при подготовке тестовых примеров следует классифицировать их по подпроблемам предметной области, выделяя стандартные случаи, определяя границы трудных ситуаций и т.п.

Ввод-вывод характеризуется данными, приобретенными в ходе диалога с экспертом, и заключениями, предъявленными ЭС в ходе объяснений. Методы приобретения данных могут не давать требуемых результатов, так как, например, задавались неправильные вопросы или собрана не вся необходимая информация. Кроме того, вопросы системы могут быть трудными для понимания, многозначными и не соответствующими знаниям пользователя. Ошибки при вводе могут возникать также из-за неудобного для пользователя входного языка. В ряде приложения для пользователя удобен ввод не только в печатной, но и в графической или звуковой форме.

Выходные сообщения (заключения) системы могут оказаться непонятны пользователю (эксперту) по разным причинам. Например, их может быть слишком много или, наоборот, слишком мало. Также причиной ошибок может являться неудачная организация, упорядоченность заключений или неподходящий пользователю уровень абстракций с непонятной ему лексикой.

Наиболее распространенный источник ошибок в рассуждениях находится в правилах вывода. Важная причина здесь часто кроется в отсутствии учета взаимозависимости сформированных правил. Другая причина заключается в ошибочности, противоречивости и неполноте используемых правил. Если неверна посылка правила, то это может привести к употреблению правила в неподходящем контексте. Если ошибочно действие правила, то трудно предсказать конечный результат. Правило может быть ошибочно, если при корректности его условия и действия нарушено соответствие между ними.

Нередко к ошибкам в работе ЭС приводят применяемые управляющие стратегии. Изменение стратегии бывает необходимо, например, если ЭС анализирует сущности в порядке, отличном от "естественного" для эксперта. Последовательность, в которой данные рассматриваются ЭС, не только влияет на эффективность работы системы, но и может приводить к изменению конечного результата. Так, рассмотрение правила А до правила В способно привести к тому, что правило В всегда будет игнорироваться системой. Изменение стратегии бывает также необходимо и в случае неэффективной работы ЭС. Кроме того, недостатки в управляющих стратегиях могут привести к чрезмерно сложным заключениям и объяснениям ЭС.

Критерии оценки ЭС зависят от точки зрения. Например, при тестировании ЭС-1 главным в оценке работы системы является полнота и безошибочность правил вывода. При тестировании промышленной системы превалирует точка зрения инженера по знаниям, которого в первую очередь интересует вопрос оптимизации представления и манипулирования знаниями. И, наконец, при тестировании ЭС после опытной эксплуатации оценка производится с точки зрения пользователя, заинтересованного в удобстве работы и получения практической пользы

Этап опытной эксплуатации

На этом этапе проверяется пригодность ЭС для конечного пользователя. Пригодность ЭС для пользователя определяется в основном удобством работы с ней и ее полезностью. Под полезностью ЭС понимается ее способность в ходе диалога определять потребности пользователя, выявлять и устранять причины неудач в работе, а также удовлетворять указанные потребности пользователя (решать поставленные задачи). В свою очередь, удобство работы с ЭС подразумевает естественность взаимодействия с ней (общение в привычном, не утомляющем пользователя виде), гибкость ЭС (способность системы настраиваться на различных пользователей, а также учитывать изменения в квалификации одного и того же пользователя) и устойчивость системы к ошибкам (способность не выходить из строя при ошибочных действиях неопытного пользователях).

В ходе разработки ЭС почти всегда осуществляется ее модификация. Выделяют следующие виды модификации системы: переформулирование понятий и требований, переконструирование представления знаний в системе и усовершенствование прототипа.

Экспертные системы, параллельные и последовательные решения

Как мы можем заметить, в большинстве алгоритмов распознавания образов подразумевается, что к началу работы алгоритма уже известна вся входная информация, которая перерабатывается параллельно. Однако ее получение зачастую требует определенных усилий. Да и наши наблюдения за реальными экспертами подтверждают, что зачастую они задают два-три вопроса, после чего делают правильные выводы. Представьте себе, если бы врач (эксперт в области медицины) перед постановкой диагноза "ангина" заставлял бы пациента пройти полное обследование вплоть до колоноскопии и пункции позвоночника (я не пробовал ни то, ни другое, но думаю, что это малоприятные вещи, а также значительная потеря времени).

Соответственно большинство алгоритмов модифицируются, чтобы обеспечить выполнение следующих условий:

* алгоритмы должны работать в условиях неполной информации (последовательно);
* последовательность запроса информации должна быть оптимальна по критериям быстроты получения результата и (или) наименьшей трудоемкости (болезненности, стоимости и т.д.) получения этой информации.

Одной из возможных стратегий для оптимизирования запросов является стратегия получения в первую очередь той информации, которая подтверждает либо опровергает наиболее вероятный на текущий момент результат. Другими словами мы пытаемся подтвердить или опровергнуть наши догадки (обратный вывод).

Пример ЭС, основанной на правилах логического вывода и действующую в обратном порядке

Допустим, вы хотите построить ЭС в области медицинской диагностики. В этом случае вам вряд ли нужно строить систему, использующую обучение на примерах, потому что имеется большое количество доступной информации, позволяющей непосредственно решать такие проблемы. К сожалению, эта информация приведена в неподходящем для обработки на компьютере виде.

Возьмите медицинскую энциклопедию и найдите в ней статью, например, о гриппе. Вы обнаружите, что в ней приведены все симптомы, причем они бесспорны. Другими словами, при наличии указанных симптомов всегда можно поставить точный диагноз.

Но чтобы использовать информацию, представленную в таком виде, вы должны обследовать пациента, решить, что у него грипп, а потом заглянуть в энциклопедию, чтобы убедиться, что у него соответствующие симптомы. Что-то здесь не так. Ведь необходимо, чтобы вы могли обследовать пациента, решить, какие у него симптомы, а потом по этим симптомам определить, чем он болен. Энциклопедия же, похоже, не позволяет сделать это так, как надо. Нам нужна не болезнь со множеством симптомов, а система, представляющая группу симптомов с последующим названием болезни. Именно это мы сейчас и попробуем сделать.

Идеальной будет такая ситуация, при которой мы сможем в той или иной области предоставить машине в приемлемом для нее виде множество определений, которые она сможет использовать примерно так же, как человек-эксперт. Именно это и пытаются делать такие программы, как PUFF, DENDRAL, PROSPECTOR.

С учетом байесовской системы логического вывода примем, что большая часть информации не является абсолютно точной, а носит вероятностный характер. Итак, начнем программирование:

|  |  |
| --- | --- |
| № | Симптомы |
| 1 | Симптом\_1 |
| 2 | Симптом\_2 |
| N | Симптом\_N |

Полученный формат данных мы будем использовать для хранения симптомов. При слове "симптомы" создается впечатление, что мы связаны исключительно с медициной, хотя речь может идти о чем угодно. Суть в том, что компьютер задает множество вопросов, содержащихся в виде символьных строк <Симптом\_1>, <Симптом\_2> и т.д.

Например, Симптом\_1 может означать строку "Много ли вы кашляете?", или, если вы пытаетесь отремонтировать неисправный автомобиль, — строку "Ослаб ли свет фар?".

Теперь оформим болезни:

|  |  |  |  |
| --- | --- | --- | --- |
| № | Болезнь | p | [j, py, pn] |
| 1 | Болезнь\_1 | p1 | [j, py, pn]1 |
| 2 | Болезнь\_2 | p2 | [j, py, pn]2 |
| N | Болезнь\_N | pn | [j, py, pn]n |

В таком виде мы будем хранить информацию о болезнях. Это не обязательно должны быть болезни — могут быть любые результаты, и каждый оператор содержит один возможный исход и всю информацию, относящуюся к нему.

Поле "болезнь" характеризует название возможного исхода, например "Грипп". Следующее поле — p — это априорная вероятность такого исхода P(H), т.е. вероятность исхода в случае отсутствия дополнительной информации. После этого идет ряд повторяющихся полей из трех элементов. Первый элемент — j — это номер соответствующего симптома (свидетельства, переменной, вопроса, если вы хотите назвать его по-другому). Следующие два элемента — P(E : H) и P(E : не H) — соответственно вероятности получения ответа "Да" на этот вопрос, если возможный исход верен и неверен. Например:

|  |  |  |  |
| --- | --- | --- | --- |
| 2010 | Грипп | 0.01 | (1, 0.9, 0.01); (2, 1, 0.01); (3, 0, 0.01) |

Здесь сказано существует априорная вероятность P(H)=0.01, что любой наугад взятый человек болеет гриппом.

Допустим, программа задает вопрос 1 (симптом 1). Тогда мы имеем P(E : H)=0.9 и P(E : не H)=0.01, а это означает, что если у пациента грипп, то он в девяти случаях из десяти ответит "да" на этот вопрос, а если у него нет гриппа, он ответит "да" лишь в одном случае из ста. Очевидно, ответ "да" подтверждает гипотезу о том, что у него грипп. Ответ "нет" позволяет предположить, что человек гриппом не болеет.

Так же и во второй группе симптомов (2, 1, 0.01). В этом случае P(E : H)=0.9, т.е. если у человека грипп, то этот симптом должен присутствовать. Соответствующий симптом может существовать и при отсутствии гриппа ( P(E : не H)=0.01 ), но это маловероятно.

Вопрос 3 исключает грипп при ответе "да", потому что P(E : H)=0. Это может быть вопрос вроде такого: "наблюдаете ли вы такой симптом на протяжении большей части жизни?" — или что-нибудь вроде этого.

Нужно подумать, — а если вы хотите получить хорошие результаты, то и провести исследование, — чтобы установить обоснованные значения для этих вероятностей. И если быть честным, то получение такой информации — вероятно, труднейшая задача, в решении которой компьютер также сможет существенно помочь Вам. Если вы напишите программу общего назначения, ее основой будет теорема Байеса, утверждающая:

P(H : E) = P(E : H) \* P(H) / (P(E : H) \* P(H) +P(E : не H) \* P(не H).

Вероятность осуществления некой гипотезы H при наличии определенных подтверждающих свидетельств Е вычисляется на основе априорной вероятности этой гипотезы без подтверждающих свидетельств и вероятности осуществления свидетельств при условиях, что гипотеза верна или неверна.

Поэтому, возвращаясь к нашим болезням, увидим:

P(H : E) = py \* p / (py \* p + pn \* (1 - p)).

В данном случае мы начинаем с того, что Р(Н) = р для всех болезней. Программа задает соответствующий вопрос и в зависимости от ответа вычисляет P(H : E). Ответ "да" подтверждает вышеуказанные расчеты, ответ "нет" тоже, но с (1 – py) вместо py и (1 – pn) вместо pn. Сделав так, мы забываем об этом, за исключением того, что априорная вероятность P(H) заменяется на P(H : E). Затем продолжается выполнение программы, но с учетом постоянной коррекции значения P(H) по мере поступления новой информации.

Описывая алгоритм, мы можем разделить программу на несколько частей.

Часть 1.

Ввод данных.

Часть 2.

Просмотр данных на предмет нахождения априорной вероятности P(H). Программа вырабатывает некоторые значения массива правил и размещает их в массиве RULEVALUE. Это делается для того, чтобы определить, какие вопросы (симптомы) являются самыми важными, и выяснить, о чем спрашивать в первую очередь. Если вы вычислите для каждого вопроса RULEVALUE[I] = RULEVALUE[I] + ABS (P(H : E) – P(H : не E)), то получите значения возможных изменений вероятностей всех болезней, к которым они относятся.

Часть 3.

Программа находит самый важный вопрос и задает его. Существует ряд вариантов, что делать с ответом: вы можете просто сказать: "да" или "нет". Можете попробовать сказать "не знаю", — изменений при этом не произойдет. Гораздо сложнее использовать шкалу от –5 до +5, чтобы выразить степень уверенности в ответе.

Часть 4.

Априорные вероятности заменяются новыми значениями при получении новых подтверждающих свидетельств.

Часть 5.

Подсчитываются новые значения правил. Определяются также минимальное и максимальное значения для каждой болезни, основанные на существующих в данный момент априорных вероятностях и предположениях, что оставшиеся свидетельства будут говорить в пользу гипотезы или противоречить ей. Важно выяснить: стоит ли данную гипотезу продолжать рассматривать или нет? Гипотезы, которые не имеют смысла, просто отбрасываются. Те же из них, чьи минимальные значения выше определенного уровня, могут считаться возможными исходами. После этого возвращаемся к части 3.

* Просмотр
* [Редактировать](https://moodle.bgpu.ru/mod/lesson/edit.php?id=71573)
* [Отчеты](https://moodle.bgpu.ru/mod/lesson/report.php?id=71573)
* [Оценить эссе](https://moodle.bgpu.ru/mod/lesson/essay.php?id=71573)

Экспертные системы

Экспертные системы, базовые понятия

Об экспертных системах (ЭС) можно говорить много и сложно. Но наш разговор очень упростится, если мы будем исходить из следующего определения экспертной системы. Экспертная система — это программа (на современном уровне развития человечества), которая заменяет эксперта в той или иной области.

Отсюда вытекает простой вывод — все, что мы изучаем в курсе "Основы проектирования систем с ИИ", конечной целью ставит разработку ЭС. В этой лекции мы остановимся только на некоторых особенностях их построения, которые не затрагиваются в остальных лекциях.

ЭС предназначены, главным образом, для решения практических задач, возникающих в слабо структурированной и трудно формализуемой предметной области. ЭС были первыми системами, которые привлекли внимание потенциальных потребителей продукции искусственного интеллекта.

С ЭС связаны некоторые распространенные заблуждения.

Заблуждение первое: ЭС будут делать не более (а скорее даже менее) того, чем может эксперт, создавший данную систему. Для опровержения данного постулата можно построить самообучающуюся ЭС в области, в которой вообще нет экспертов, либо объединить в одной ЭС знания нескольких экспертов, и получить в результате систему, которая может то, чего ни один из ее создателей не может.

Заблуждение второе: ЭС никогда не заменит человека-эксперта. Уже заменяет, иначе зачем бы их создавали?

Экспертные системы, методика построения

В настоящее время сложилась определенная технология разработки ЭС, которая включает следующие шесть этапов: идентификация, концептуализация, формализация, выполнение, тестирование и опытная эксплуатация.

Рис. 9.1. Методика (этапы) разработки ЭС

Этап идентификации

Этап идентификации связан, прежде всего, с осмыслением тех задач, которые предстоит решить будущей ЭС, и формированием требований к ней. Результатом данного этапа является ответ на вопрос, что надо сделать и какие ресурсы необходимо задействовать (идентификация задачи, определение участников процесса проектирования и их роли, выявление ресурсов и целей).

Обычно в разработке ЭС участвуют не менее трех-четырех человек — один эксперт, один или два инженера по знаниям и один программист, привлекаемый для модификации и согласования инструментальных средств. Также к процессу разработки ЭС могут по мере необходимости привлекаться и другие участники. Например, инженер по знаниям может пригласить других экспертов, чтобы убедиться в правильности своего понимания основного эксперта, представительности тестов, демонстрирующих особенности рассматриваемой задачи, совпадения взглядов различных экспертов на качество предлагаемых решений. Кроме того, для сложных систем считается целесообразным привлекать к основному циклу разработки несколько экспертов. Однако в этом случае, как правило, требуется, чтобы один из экспертов отвечал за непротиворечивость знаний, сообщаемых коллективом экспертов.

Идентификация задачи заключается в составлении неформального (вербального) описания, в котором указываются: общие характеристики задачи; подзадачи, выделяемые внутри данной задачи; ключевые понятия (объекты), их входные (выходные) данные; предположительный вид решения, а также знания, относящиеся к решаемой задаче.

В процессе идентификации задачи инженер по знаниям и эксперт работают в тесном контакте. Начальное неформальное описание задачи экспертом используется инженером по знаниям для уточнения терминов и ключевых понятий. Эксперт корректирует описание задачи, объясняет, как решать ее и какие рассуждения лежат в основе того или иного решения. После нескольких циклов, уточняющих описание, эксперт и инженер по знаниям получают окончательное неформальное описание задачи.

При проектировании ЭС типичными ресурсами являются источники знаний, время разработки, вычислительные средства и объем финансирования. Для эксперта источниками знаний служат его предшествующий опыт по решению задачи, книги, известные примеры решения задач, а для инженера по знаниям — опыт в решении аналогичных задач, методы представления знаний и манипулирования ими, программные инструментальные средства. При определении времени разработки обычно имеется в виду, что сроки разработки и внедрения ЭС составляют, как правило, не менее года (при трудоемкости 5 чел.-лет). Определение объема финансирования оказывает существенное влияние на процесс разработки, так как, например, при недостаточном финансировании предпочтение может быть отдано не разработке оригинальной новой системы, а адаптации существующей.

При идентификации целей важно отличать цели, ради которых создается ЭС, от задач, которые она должна решать. Примерами возможных целей являются: формализация неформальных знаний экспертов; улучшение качества решений, принимаемых экспертом; автоматизация рутинных аспектов работы эксперта (пользователя); тиражирование знаний эксперта.

Этап концептуализации

На данном этапе проводится содержательный анализ проблемной области, выявляются используемые понятия и их взаимосвязи, определяются методы решения задач. Этот этап завершается созданием модели предметной области (ПО), включающей основные концепты и отношения. На этапе концептуализации определяются следующие особенности задачи: типы доступных данных; исходные и выводимые данные, подзадачи общей задачи; применяемые стратегии и гипотезы; виды взаимосвязей между объектами ПО, типы используемых отношений (иерархия, причина — следствие, часть — целое и т.п.); процессы, применяемые в ходе решения; состав знаний, используемых при решении задачи; типы ограничений, накладываемых на процессы, которые применены в ходе решения; состав знаний, используемых для обоснования решений.

Существует два подхода к процессу построения модели предметной области, которая является целью разработчиков ЭС на этапе концептуализации. Признаковый или атрибутивный подход предполагает наличие полученной от экспертов информации в виде троек объект—атрибут—значение атрибута, а также наличие обучающей информации. Этот подход развивается в рамках направления, получившего название "формирование знаний" или "машинное обучение" (machine learning).

Второй подход, называемый структурным (или когнитивным), осуществляется путем выделения элементов предметной области, их взаимосвязей и семантических отношений.

Для атрибутивного подхода характерно наличие наиболее полной информации о предметной области: об объектах, их атрибутах и о значениях атрибутов. Кроме того, существенным моментом является использование дополнительной обучающей информации, которая задается группированием объектов в классы по тому или иному содержательному критерию. Тройки объект—атрибут—значение атрибута могут быть получены с помощью так называемого метода реклассификации, который основан на предположении что [задача](https://moodle.bgpu.ru/mod/assign/view.php?id=109703) является объектно-ориентированной и объекты задачи хорошо известны эксперту. Идея метода состоит в том, что конструируются правила (комбинации значений атрибутов), позволяющие отличить один объект от другого. Обучающая информация может быть задана на основании прецедентов правильных экспертных заключений, например, с помощью метода извлечения знаний, получившего название "анализ протоколов мыслей вслух".

При наличии обучающей информации для формирования модели предметной области на этапе концептуализации можно использовать весь арсенал методов, развиваемых в рамках задачи распознавания образов. Таким образом, несмотря на то, что здесь атрибутивному подходу не уделено много места, он является одним из потребителей всего того, что было указано в лекции, посвященной распознаванию образов и автоматического группирования данных.

Структурный подход к построению модели предметной области предполагает выделение следующих когнитивных элементов знаний: 1. Понятия. 2. Взаимосвязи. 3. Метапонятия. 4. Семантические отношения.

Выделяемые понятия предметной области должны образовывать систему, под которой понимается совокупность понятий, обладающая следующими свойствами: уникальностью (отсутствием избыточности); полнотой (достаточно полным описанием различных процессов, фактов, явлений и т.д. предметной области); достоверностью (валидностью — соответствием выделенных единиц смысловой информации их реальным наименованиям) и непротиворечивостью (отсутствием омонимии).

При построении системы понятий с помощью "метода локального представления" эксперта просят разбить задачу на подзадачи для перечисления целевых состояний и описания общих категорий цели. Далее для каждого разбиения (локального представления) эксперт формулирует информационные факты и дает им четкое наименование (название). Считается, что для успешного решения задачи построения модели предметной области число таких информационных фактов в каждом локальном представлении, которыми человек способен одновременно манипулировать, должно быть примерно равно семи.

"Метод вычисления коэффициента использования" основан на следующей гипотезе. Элемент данных (или информационный факт) может являться понятием, если он:

* используется в большом числе подзадач;
* используется с большим числом других элементов данных;
* редко используется совместно с другими элементами данных по сравнению с общим числом его применения во всех подзадачах (это и есть коэффициент использования).

Полученные значения могут служить критерием для классификации всех элементов данных и, таким образом, для формирования системы понятий.

"Метод формирования перечня понятий" заключается в том, что экспертам (желательно, чтобы их было больше двух) дается [задание](https://moodler2.bgpu.ru/mod/assign/view.php?id=109840) составить список понятий, относящихся к исследуемой предметной области. Понятия, выделенные всеми экспертами, включаются в систему понятий, остальные подлежат обсуждению.

"Ролевой метод" состоит в том, что эксперту дается [задание](https://moodler2.bgpu.ru/mod/assign/view.php?id=109840) обучить инженера по знаниям решению некоторых задач предметной области. Таким образом, эксперт играет роль учителя, а инженер по знаниям — роль ученика. Процесс обучения записывается на магнитофон. Затем третий участник прослушивает магнитофонную ленту и выписывает на бумаге все понятия, употребленные учителем или учеником.

При использовании метода "составления списка элементарных действий" эксперту дается [задание](https://moodler2.bgpu.ru/mod/assign/view.php?id=109840) составить такой список при решении задачи в произвольном порядке.

В методе "составление оглавления учебника" эксперту предлагается представить ситуацию, в которой его попросили написать учебник. Необходимо составить на бумаге перечень предполагаемых глав, разделов, параграфов, пунктов и подпунктов книги.

"Текстологический метод" формирования системы понятий заключается в том, что эксперту дается [задание](https://moodler2.bgpu.ru/mod/assign/view.php?id=109840) выписать из руководств (книг по специальности) некоторые элементы, представляющие собой единицы смысловой информации.

Группа методов установления взаимосвязей предполагает установление семантической близости между отдельными понятиями. В основе установления взаимосвязей лежит психологический эффект "свободных ассоциаций", а также фундаментальная категория близости объектов или концептов.

Эффект свободных ассоциаций заключается в следующем. Испытуемого просят отвечать на заданное слово первым пришедшим на ум словом. Как правило, реакция большинства испытуемых (если слова не были слишком необычными) оказывается одинаковой. Количество переходов в цепочке может служить мерой "смыслового расстояния" между двумя понятиями. Многочисленные опыты подтверждают гипотезу, что для двух любых слов (понятий) существует ассоциативная цепочка, состоящая не более чем из семи слов.

"Метод свободных ассоциаций" основан на психологическом эффекте, описанном выше. Эксперту предъявляется понятие с просьбой назвать как можно быстрее первое пришедшее на ум понятие из сформированной ранее системы понятий. Далее производится анализ полученной информации.

В методе "сортировка карточек" исходным материалом служат выписанные на карточки понятия. Применяются два варианта метода. В первом эксперту задаются некоторые глобальные критерии предметной области, которыми он должен руководствоваться при раскладывании карточек на группы. Во втором случае, когда сформулировать глобальные критерии невозможно, эксперту дается [задание](https://moodler2.bgpu.ru/mod/assign/view.php?id=109840) разложить карточки на группы в соответствии с интуитивным пониманием семантической близости предъявляемых понятий.

"Метод обнаружения регулярностей" основан на гипотезе о том, что элементы цепочки понятия, которые человек вспоминает с определенной регулярностью, имеют тесную ассоциативную взаимосвязь. Для эксперимента произвольным образом отбирается 20 понятий. Эксперту предъявляется одно из числа отобранных. Процедура повторяется до 20 раз, причем каждый раз начальные концепты должны быть разными. Затем инженер по знаниям анализирует полученные цепочки с целью нахождения постоянно повторяющихся понятий (регулярностей). Внутри выделенных таким образом группировок устанавливаются ассоциативные взаимосвязи.

Кроме рассмотренных выше неформальных методов для установления взаимосвязей между отдельными понятиями применяются также формальные методы. Сюда в первую очередь относятся методы семантического дифференциала и репертуарных решеток.

Выделенные понятия предметной области и установленные между ними взаимосвязи служат основанием для дальнейшего построения системы метапонятий — осмысленных в контексте изучаемой предметной области системы группировок понятий. Для определения этих группировок применяют как неформальные, так и формальные методы.

Интерпретация, как правило, легче дается эксперту, если группировки получены неформальными методами. В этом случае выделенные классы более понятны эксперту. Причем в некоторых предметных областях совсем не обязательно устанавливать взаимосвязи между понятиями, так как метапонятия, образно говоря, "лежат на поверхности".

Последним этапом построения модели предметной области при концептуальном анализе является установление семантических отношений между выделенными понятиями и метапонятиями. Установить семантические отношения — это значит определить специфику взаимосвязи, полученной в результате применения тех или иных методов. Для этого необходимо каждую зафиксированную взаимосвязь осмыслить и отнести ее к тому или иному типу отношений.

Существует около 200 базовых отношений, например, "часть — целое", "род — вид", "причина — следствие", пространственные, временные и другие отношения. Для каждой предметной области помимо общих базовых отношений могут существовать и уникальные отношения.

"Прямой метод" установления семантических отношений основан на непосредственном осмыслении каждой взаимосвязи. В том случае, когда эксперт затрудняется дать интерпретацию выделенной взаимосвязи, ему предлагается следующая процедура. Формируются тройки: понятие 1 — связь — понятие 2. Рядом с каждой тройкой записывается короткое предложение или фраза, построенное так, чтобы понятие 1 и понятие 2 входили в это предложение. В качестве связок используются только содержательные отношения и не применяются неопределенные связки типа "похож на" или "связан с".

Для "косвенного метода" не обязательно иметь взаимосвязи, достаточно лишь наличие системы понятий. Формулируется некоторый критерий, для которого из системы понятий выбирается определенная совокупность концептов. Эта совокупность предъявляется эксперту с просьбой дать вербальное описание сформулированного критерия. Концепты предъявляются эксперту все сразу (желательно на карточках). В случае затруднений эксперта прибегают к разбиению отобранных концептов на группы с помощью более мелких критериев. Исходное количество концептов может быть произвольным, но после разбиения на группы в каждой из таких групп должно быть не более десяти концептов. После того как составлены описания по всем группам, эксперту предлагают объединить эти описания в одно.

Следующий шаг в косвенном методе установления семантических отношений — это анализ текста, составленного экспертом. Концепты заменяют цифрами (это может быть исходная нумерация), а связки оставляют. Тем самым строится некоторый граф, вершинами которого служат концепты, а дугами — связки (например, "ввиду", "приводит к", "выражаясь с одной стороны", "обусловливая", "сочетаясь", "определяет", "вплоть до" и т.д.) Этот метод позволяет устанавливать не только базовые отношения, но и отношения, специфические для конкретной предметной области.

Рассмотренные выше методы формирования системы понятий и метапонятий, установления взаимосвязей и семантических отношений в разных сочетаниях применяются на этапе концептуализации при построении модели предметной области.

Этап формализации

Теперь все ключевые понятия и отношения выражаются на некотором формальном языке, который либо выбирается из числа уже существующих, либо создается заново. Другими словами, на данном этапе определяются состав средств и способы представления декларативных и процедурных знаний, осуществляется это представление и в итоге формируется описание решения задачи ЭС на предложенном (инженером по знаниям) формальном языке.

Выходом этапа формализации является описание того, как рассматриваемая задача может быть представлена в выбранном или разработанном формализме. Сюда относится указание способов представления знаний (фреймы, сценарии, семантические сети и т.д.) и определение способов манипулирования этими знаниями (логический вывод, аналитическая модель, статистическая модель и др.) и интерпретации знаний.

Этап выполнения

Цель этого этапа — создание одного или нескольких прототипов ЭС, решающих требуемые задачи. Затем на данном этапе по результатам тестирования и опытной эксплуатации создается конечный продукт, пригодный для промышленного использования. Разработка прототипа состоит в программировании его компонентов или выборе их из известных инструментальных средств и наполнении базы знаний.

Главное в создании прототипа заключается в том, чтобы этот прототип обеспечил проверку адекватности идей, методов и способов представления знаний решаемым задачам. Создание первого прототипа должно подтвердить, что выбранные методы решений и способы представления пригодны для успешного решения, по крайней мере, ряда задач из актуальной предметной области, а также продемонстрировать тенденцию к получению высококачественных и эффективных решений для всех задач предметной области по мере увеличения объема знаний.

После разработки первого прототипа ЭС-1 круг предлагаемых для решения задач расширяется, и собираются пожелания и замечания, которые должны быть учтены в очередной версии системы ЭС-2. Осуществляется развитие ЭС-1 путем добавления "дружественного" интерфейса, средств для исследования базы знаний и цепочек выводов, генерируемых системой, а также средств для сбора замечаний пользователей и средств хранения библиотеки задач, решенных системой.

Выполнение экспериментов с расширенной версией ЭС-1, анализ пожеланий и замечаний служат отправной точкой для создания второго прототипа ЭС-2. Процесс разработки ЭС-2 — итеративный. Он может продолжаться от нескольких месяцев до нескольких лет в зависимости от сложности предметной области, гибкости выбранного представления знаний и степени соответствия управляющего механизма решаемым задачам (возможно, потребуется разработка ЭС-3 и т.д.). При разработке ЭС-2, кроме перечисленных задач, решаются следующие:

* анализ функционирования системы при значительном расширении базы знаний;
* исследование возможностей системы в решении более широкого круга задач и принятие мер для обеспечения таких возможностей;
* анализ мнений пользователей о функционировании ЭС;
* разработка системы ввода-вывода, осуществляющей анализ или синтез предложений ограниченного естественного языка, позволяющей взаимодействовать с ЭС-2 в форме, близкой к форме стандартных учебников для данной области.

Если ЭС-2 успешно прошла этап тестирования, то она может классифицироваться как промышленная экспертная система.

Этап тестирования

В ходе данного этапа производится оценка выбранного способа представления знаний в ЭС в целом. Для этого инженер по знаниям подбирает примеры, обеспечивающие проверку всех возможностей разработанной ЭС.

Различают следующие источники неудач в работе системы: тестовые примеры, ввод-вывод, правила вывода, управляющие стратегии.

Показательные тестовые примеры являются наиболее очевидной причиной неудачной работы ЭС. В худшем случае тестовые примеры могут оказаться вообще вне предметной области, на которую рассчитана ЭС, однако чаще множество тестовых примеров оказывается слишком однородным и не охватывает всю предметную область. Поэтому при подготовке тестовых примеров следует классифицировать их по подпроблемам предметной области, выделяя стандартные случаи, определяя границы трудных ситуаций и т.п.

Ввод-вывод характеризуется данными, приобретенными в ходе диалога с экспертом, и заключениями, предъявленными ЭС в ходе объяснений. Методы приобретения данных могут не давать требуемых результатов, так как, например, задавались неправильные вопросы или собрана не вся необходимая информация. Кроме того, вопросы системы могут быть трудными для понимания, многозначными и не соответствующими знаниям пользователя. Ошибки при вводе могут возникать также из-за неудобного для пользователя входного языка. В ряде приложения для пользователя удобен ввод не только в печатной, но и в графической или звуковой форме.

Выходные сообщения (заключения) системы могут оказаться непонятны пользователю (эксперту) по разным причинам. Например, их может быть слишком много или, наоборот, слишком мало. Также причиной ошибок может являться неудачная организация, упорядоченность заключений или неподходящий пользователю уровень абстракций с непонятной ему лексикой.

Наиболее распространенный источник ошибок в рассуждениях находится в правилах вывода. Важная причина здесь часто кроется в отсутствии учета взаимозависимости сформированных правил. Другая причина заключается в ошибочности, противоречивости и неполноте используемых правил. Если неверна посылка правила, то это может привести к употреблению правила в неподходящем контексте. Если ошибочно действие правила, то трудно предсказать конечный результат. Правило может быть ошибочно, если при корректности его условия и действия нарушено соответствие между ними.

Нередко к ошибкам в работе ЭС приводят применяемые управляющие стратегии. Изменение стратегии бывает необходимо, например, если ЭС анализирует сущности в порядке, отличном от "естественного" для эксперта. Последовательность, в которой данные рассматриваются ЭС, не только влияет на эффективность работы системы, но и может приводить к изменению конечного результата. Так, рассмотрение правила А до правила В способно привести к тому, что правило В всегда будет игнорироваться системой. Изменение стратегии бывает также необходимо и в случае неэффективной работы ЭС. Кроме того, недостатки в управляющих стратегиях могут привести к чрезмерно сложным заключениям и объяснениям ЭС.

Критерии оценки ЭС зависят от точки зрения. Например, при тестировании ЭС-1 главным в оценке работы системы является полнота и безошибочность правил вывода. При тестировании промышленной системы превалирует точка зрения инженера по знаниям, которого в первую очередь интересует вопрос оптимизации представления и манипулирования знаниями. И, наконец, при тестировании ЭС после опытной эксплуатации оценка производится с точки зрения пользователя, заинтересованного в удобстве работы и получения практической пользы

Этап опытной эксплуатации

На этом этапе проверяется пригодность ЭС для конечного пользователя. Пригодность ЭС для пользователя определяется в основном удобством работы с ней и ее полезностью. Под полезностью ЭС понимается ее способность в ходе диалога определять потребности пользователя, выявлять и устранять причины неудач в работе, а также удовлетворять указанные потребности пользователя (решать поставленные задачи). В свою очередь, удобство работы с ЭС подразумевает естественность взаимодействия с ней (общение в привычном, не утомляющем пользователя виде), гибкость ЭС (способность системы настраиваться на различных пользователей, а также учитывать изменения в квалификации одного и того же пользователя) и устойчивость системы к ошибкам (способность не выходить из строя при ошибочных действиях неопытного пользователях).

В ходе разработки ЭС почти всегда осуществляется ее модификация. Выделяют следующие виды модификации системы: переформулирование понятий и требований, переконструирование представления знаний в системе и усовершенствование прототипа.

Экспертные системы, параллельные и последовательные решения

Как мы можем заметить, в большинстве алгоритмов распознавания образов подразумевается, что к началу работы алгоритма уже известна вся входная информация, которая перерабатывается параллельно. Однако ее получение зачастую требует определенных усилий. Да и наши наблюдения за реальными экспертами подтверждают, что зачастую они задают два-три вопроса, после чего делают правильные выводы. Представьте себе, если бы врач (эксперт в области медицины) перед постановкой диагноза "ангина" заставлял бы пациента пройти полное обследование вплоть до колоноскопии и пункции позвоночника (я не пробовал ни то, ни другое, но думаю, что это малоприятные вещи, а также значительная потеря времени).

Соответственно большинство алгоритмов модифицируются, чтобы обеспечить выполнение следующих условий:

* алгоритмы должны работать в условиях неполной информации (последовательно);
* последовательность запроса информации должна быть оптимальна по критериям быстроты получения результата и (или) наименьшей трудоемкости (болезненности, стоимости и т.д.) получения этой информации.

Одной из возможных стратегий для оптимизирования запросов является стратегия получения в первую очередь той информации, которая подтверждает либо опровергает наиболее вероятный на текущий момент результат. Другими словами мы пытаемся подтвердить или опровергнуть наши догадки (обратный вывод).

Пример ЭС, основанной на правилах логического вывода и действующую в обратном порядке

Допустим, вы хотите построить ЭС в области медицинской диагностики. В этом случае вам вряд ли нужно строить систему, использующую обучение на примерах, потому что имеется большое количество доступной информации, позволяющей непосредственно решать такие проблемы. К сожалению, эта информация приведена в неподходящем для обработки на компьютере виде.

Возьмите медицинскую энциклопедию и найдите в ней статью, например, о гриппе. Вы обнаружите, что в ней приведены все симптомы, причем они бесспорны. Другими словами, при наличии указанных симптомов всегда можно поставить точный диагноз.

Но чтобы использовать информацию, представленную в таком виде, вы должны обследовать пациента, решить, что у него грипп, а потом заглянуть в энциклопедию, чтобы убедиться, что у него соответствующие симптомы. Что-то здесь не так. Ведь необходимо, чтобы вы могли обследовать пациента, решить, какие у него симптомы, а потом по этим симптомам определить, чем он болен. Энциклопедия же, похоже, не позволяет сделать это так, как надо. Нам нужна не болезнь со множеством симптомов, а система, представляющая группу симптомов с последующим названием болезни. Именно это мы сейчас и попробуем сделать.

Идеальной будет такая ситуация, при которой мы сможем в той или иной области предоставить машине в приемлемом для нее виде множество определений, которые она сможет использовать примерно так же, как человек-эксперт. Именно это и пытаются делать такие программы, как PUFF, DENDRAL, PROSPECTOR.

С учетом байесовской системы логического вывода примем, что большая часть информации не является абсолютно точной, а носит вероятностный характер. Итак, начнем программирование:

|  |  |
| --- | --- |
| № | Симптомы |
| 1 | Симптом\_1 |
| 2 | Симптом\_2 |
| N | Симптом\_N |

Полученный формат данных мы будем использовать для хранения симптомов. При слове "симптомы" создается впечатление, что мы связаны исключительно с медициной, хотя речь может идти о чем угодно. Суть в том, что компьютер задает множество вопросов, содержащихся в виде символьных строк <Симптом\_1>, <Симптом\_2> и т.д.

Например, Симптом\_1 может означать строку "Много ли вы кашляете?", или, если вы пытаетесь отремонтировать неисправный автомобиль, — строку "Ослаб ли свет фар?".

Теперь оформим болезни:

|  |  |  |  |
| --- | --- | --- | --- |
| № | Болезнь | p | [j, py, pn] |
| 1 | Болезнь\_1 | p1 | [j, py, pn]1 |
| 2 | Болезнь\_2 | p2 | [j, py, pn]2 |
| N | Болезнь\_N | pn | [j, py, pn]n |

В таком виде мы будем хранить информацию о болезнях. Это не обязательно должны быть болезни — могут быть любые результаты, и каждый оператор содержит один возможный исход и всю информацию, относящуюся к нему.

Поле "болезнь" характеризует название возможного исхода, например "Грипп". Следующее поле — p — это априорная вероятность такого исхода P(H), т.е. вероятность исхода в случае отсутствия дополнительной информации. После этого идет ряд повторяющихся полей из трех элементов. Первый элемент — j — это номер соответствующего симптома (свидетельства, переменной, вопроса, если вы хотите назвать его по-другому). Следующие два элемента — P(E : H) и P(E : не H) — соответственно вероятности получения ответа "Да" на этот вопрос, если возможный исход верен и неверен. Например:

|  |  |  |  |
| --- | --- | --- | --- |
| 2010 | Грипп | 0.01 | (1, 0.9, 0.01); (2, 1, 0.01); (3, 0, 0.01) |

Здесь сказано существует априорная вероятность P(H)=0.01, что любой наугад взятый человек болеет гриппом.

Допустим, программа задает вопрос 1 (симптом 1). Тогда мы имеем P(E : H)=0.9 и P(E : не H)=0.01, а это означает, что если у пациента грипп, то он в девяти случаях из десяти ответит "да" на этот вопрос, а если у него нет гриппа, он ответит "да" лишь в одном случае из ста. Очевидно, ответ "да" подтверждает гипотезу о том, что у него грипп. Ответ "нет" позволяет предположить, что человек гриппом не болеет.

Так же и во второй группе симптомов (2, 1, 0.01). В этом случае P(E : H)=0.9, т.е. если у человека грипп, то этот симптом должен присутствовать. Соответствующий симптом может существовать и при отсутствии гриппа ( P(E : не H)=0.01 ), но это маловероятно.

Вопрос 3 исключает грипп при ответе "да", потому что P(E : H)=0. Это может быть вопрос вроде такого: "наблюдаете ли вы такой симптом на протяжении большей части жизни?" — или что-нибудь вроде этого.

Нужно подумать, — а если вы хотите получить хорошие результаты, то и провести исследование, — чтобы установить обоснованные значения для этих вероятностей. И если быть честным, то получение такой информации — вероятно, труднейшая задача, в решении которой компьютер также сможет существенно помочь Вам. Если вы напишите программу общего назначения, ее основой будет теорема Байеса, утверждающая:

P(H : E) = P(E : H) \* P(H) / (P(E : H) \* P(H) +P(E : не H) \* P(не H).

Вероятность осуществления некой гипотезы H при наличии определенных подтверждающих свидетельств Е вычисляется на основе априорной вероятности этой гипотезы без подтверждающих свидетельств и вероятности осуществления свидетельств при условиях, что гипотеза верна или неверна.

Поэтому, возвращаясь к нашим болезням, увидим:

P(H : E) = py \* p / (py \* p + pn \* (1 - p)).

В данном случае мы начинаем с того, что Р(Н) = р для всех болезней. Программа задает соответствующий вопрос и в зависимости от ответа вычисляет P(H : E). Ответ "да" подтверждает вышеуказанные расчеты, ответ "нет" тоже, но с (1 – py) вместо py и (1 – pn) вместо pn. Сделав так, мы забываем об этом, за исключением того, что априорная вероятность P(H) заменяется на P(H : E). Затем продолжается выполнение программы, но с учетом постоянной коррекции значения P(H) по мере поступления новой информации.

Описывая алгоритм, мы можем разделить программу на несколько частей.

Часть 1.

Ввод данных.

Часть 2.

Просмотр данных на предмет нахождения априорной вероятности P(H). Программа вырабатывает некоторые значения массива правил и размещает их в массиве RULEVALUE. Это делается для того, чтобы определить, какие вопросы (симптомы) являются самыми важными, и выяснить, о чем спрашивать в первую очередь. Если вы вычислите для каждого вопроса RULEVALUE[I] = RULEVALUE[I] + ABS (P(H : E) – P(H : не E)), то получите значения возможных изменений вероятностей всех болезней, к которым они относятся.

Часть 3.

Программа находит самый важный вопрос и задает его. Существует ряд вариантов, что делать с ответом: вы можете просто сказать: "да" или "нет". Можете попробовать сказать "не знаю", — изменений при этом не произойдет. Гораздо сложнее использовать шкалу от –5 до +5, чтобы выразить степень уверенности в ответе.

Часть 4.

Априорные вероятности заменяются новыми значениями при получении новых подтверждающих свидетельств.

Часть 5.

Подсчитываются новые значения правил. Определяются также минимальное и максимальное значения для каждой болезни, основанные на существующих в данный момент априорных вероятностях и предположениях, что оставшиеся свидетельства будут говорить в пользу гипотезы или противоречить ей. Важно выяснить: стоит ли данную гипотезу продолжать рассматривать или нет? Гипотезы, которые не имеют смысла, просто отбрасываются. Те же из них, чьи минимальные значения выше определенного уровня, могут считаться возможными исходами. После этого возвращаемся к части 3.