

Обучение в интеллектуальных системах.

Формы обучения

Один из центральных элементов интеллектуального поведения – способность приспосабливаться или учиться на основе опыта. Возможность обучения или адаптивного поведения интеллектуального агента соответствует его более высокому уровню способностей.

В основе обучения лежит представление о том, что результаты восприятия должны использоваться не только для осуществления действий, но и для повышения способности агента действовать в будущем.

Формы обучения. Обучение на основе наблюдений

Обучение принимает много разных форм в зависимости от характера производительного элемента, компонента, подлежащего усовершенствованию и доступной обратной связи.

В первую очередь – **контролируемое обучение**, при котором предусматривается изучение некоторой функции на примерах ее входных и выходных данных и доступна обратная связь либо от учителя, либо от среды, позволяющая получать правильные значения, относящиеся к примерам. Изучение функции с дискретными значениями называется классификацией, а изучение непрерывной функции называется регрессией (предсказанием).

Простейшая разновидность – механическое обучение, при котором предлагается пример, а интеллектуальный агент просто копирует пример и точно воспроизводит поведение. Например, можно управлять моделированием 24 часа в день, 7 дней в неделю, производить новые ситуации и желательное поведение, затем представлять эти примеры агенту. Агент был бы уже приспособленным интерпретировать ситуации спустя одну неделю после того, как начато обучение, и еще лучше делал бы это один месяц спустя.

Другая разновидность - регулирование веса или параметра. В этом случае, мы можем знать априорно, какие факторы являются важными в решении, но мы не знаем их вклад в решение. Можно подбирать весовые факторы в течение некоторого времени так, чтобы увеличивалась вероятность правильного решения. На этой технике основано обучение на основе нейронной сети.

Индуктивное обучение - процесс контролируемого обучения, в котором извлекаются важные особенности проблемы, что позволяет вывести новую ситуацию или вход. Деревья решения и нейронные сети и выполняют индукцию и могут использоваться для классификации или регрессии проблемы. Ключевой аспект индуктивных методов заключается в том, что примеры обработаны и автоматически преобразованы во внутреннюю форму, которая представляет сущность проблемы.

Форма **неконтролируемого обучения** предполагает выявление определенных закономерностей во входных данных в случае, когда не задаются соответствующие выходные данные. Объединение входных данных в кластеры или сегментация процесса продолжаются, пока агент не помещает те же самые данные в ту же самую группу на последовательных проходах по данным. Например, агент-водитель такси может изучить понятие о “днях с хорошими условиями для движения” даже не получая обозначенных подобными заголовками примеров таких дней. Для данной формы обучения часто применяются статистические методы обучения.

Форма **обучения с подкреплением** является наиболее общей. При этом агент не получает от учителя информации о правильных действиях, а должен сам делать данный вывод на основе подкрепления (вознаграждения). Например, большой штраф за столкновение с идущим впереди автомобилем может служить для агента показателем нежелательности его поведения. Задача обучения с подкреплением обычно включает подзадачу обучения тому, как функционирует среда, в которой существует агент.

Индуктивное обучение

Основная задача чисто индуктивного логического вывода, или просто **индукции**, заключается в том, чтобы на основании совокупности примеров входных и выходных данных функции f получить функцию h , которая аппроксимирует функцию f . Функция h называется гипотезой. Качественная гипотеза должна обеспечивать приемлемое обобщение, т.е. должна правильно предсказывать появление еще не полученных примеров. В этом состоит фундаментальная проблема индукции.

Известным примером (рисунки 1 и 2) индуктивного обучения является подгонка функции от одной переменной к некоторым точкам из набора данных.

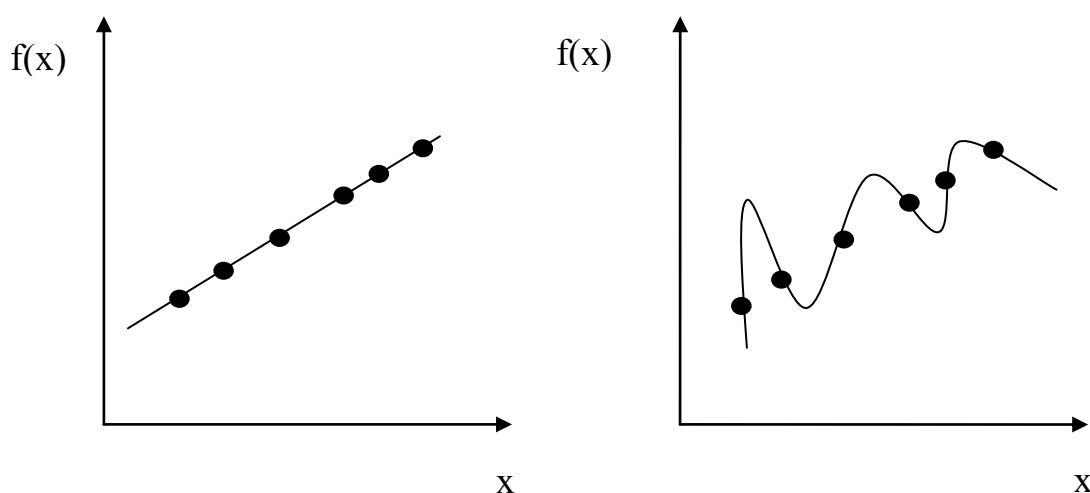


Рисунок 1 – Подгонка функции одной переменной к значениям полинома первой степени

Примеры представляют собой пары $(x, f(x))$, где x и $f(x)$ – действительные числа. Выберем в качестве пространства гипотез – множество полиномов, имеющих степень не больше k , таких как $5x^2+2$, $x^{17}-3x^3$. На рисунке 1 показаны значения, которые соответствуют некоторой прямой (полиному первой степени). Так как прямая согласуется со всеми данными, то она называется **совместимой** с гипотезой. На том же рисунке 1 показан полином более высокой степени, который также совместим с этими данными.

Это пример важной проблемы индуктивного обучения – выбору среди множества согласованных гипотез. Эта проблема может быть решена с использованием принципа “бритвы Оккама”, согласно которому предпочтение следует отдавать наиболее простой гипотезе, согласующейся с данными.

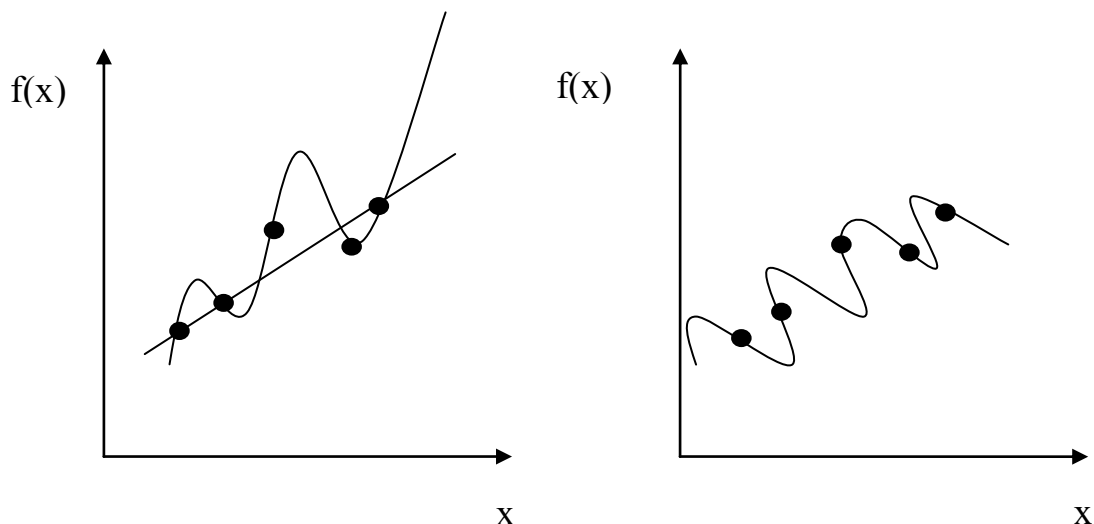


Рисунок 2 – Подгонка функции одной переменной

На рисунке 2 показан второй набор данных, с которым нельзя точно совместить прямую линию.

Для точного согласования с данным набором требуется полином четвертой степени с пятью параметрами. Возможно, что лучше согласовать этот набор данных с прямой линией, которая не будет точно совместимой, но позволит получать обоснованные предсказания. Принятие данного решения равносильно признанию недетерминированности искомой функции. При наличии недетерминированных функций неизбежно приходится искать компромисс между сложностью гипотезы и степенью ее согласованности с данными.

Возможность найти простую согласованную гипотезу зависит от выбранного пространства гипотез. На рис. 2 показано как тот же набор данных может быть точно согласован с простой функцией вида $ax+b+c\sin(x)$.

Задача обучения называется реализуемой, если Пространство гипотез содержит подходящую функцию, иначе она называется нереализуемой. В

произвольной ситуации невозможно определить, относится ли конкретная задача обучения к категории реализуемых.

Использование априорных знаний для логического вывода пространства гипотез – один из способов решения данной проблемы. Еще один подход – применение наибольшего возможного пространства гипотез, например, использование класса всех машин Тьюринга, так как известно, что любая вычислимая функция может быть представлена с помощью некоторой машины Тьюринга.

При этом приходится искать компромисс между выразительностью пространства гипотез и сложностью поиска простой, совместимой гипотезы в этом пространстве, так как неразрешима в общем виде проблема определения того, является ли конкретная машина Тьюринга совместимой с данными. Кроме того результирующие гипотезы из простых пространств гипотез могут оказаться и более простыми в использовании, т.е. вычисление $h(x)$, если h – линейная функция, будет осуществляться быстрее, чем при использовании программы, моделирующей произвольную машину Тьюринга.

Построение деревьев решений

Индуктивный логический вывод деревьев решений – одна из простейших и эффективных форм алгоритмов обучения.

Дерево решений принимает в качестве входных данных множество атрибутов, описывающих объект или ситуацию, и возвращает предсказанное выходное значение, соответствующее этим входным данным. Входные атрибуты и выходные значения могут быть дискретными или непрерывными. Будем рассматривать случай булевой классификации, когда каждый пример может быть либо истинный (положительный), либо ложный (отрицательный).

Решение, содержащееся в дереве решений, может быть получено путем выполнения последовательности проверок. В каждом узле дерева проверяет-

ся значение одного из свойств, а ветви, исходящие из этого узла, обозначаются возможными значениями результатов проверки. При достижении листового узла возвращается соответствующее ему значение.

Рассмотрим пример применения метода обучения на основе дерева решений к задаче, в которой клиент ждет, пока освободится место за столиком в ресторане.

Целевой предикат `WillWait` (Следует ли ждать). Вначале определим, какие атрибуты доступны для описания примеров ситуаций в данной проблемной области:

`Alternate` – есть ли поблизости ресторан такого же класса,
`Bar` – имеется ли в ресторане бар, в котором можно подождать,
`Fri/Sat` – атрибут истинен по пятницам и субботам,
`Hungry` – испытывает ли посетитель чувство голода,
`Patrons` – `Full/Some/None` – количество людей в ресторане,
`Price` – `*/**/**` – ценовая категория ресторана,
`Raining` – идет ли дождь на улице,
`Reservation` – забронировано ли за посетителем место,
`Type` – `French/Italian/Thai/Burger` – тип кухни в ресторане,
`WaitEstimate` – `0-10/10-30/30-60/>60` минут – время ожидания.

Возможное дерево решений показано на рисунке 3. В нем не используются атрибуты `Price` и `Type`, что означает, что лицо, принимающее решение, рассматривает их как малозначительные. Обработка примеров ситуаций с помощью этого дерева начинается от корня и проходит по соответствующей ветви до тех пор, пока не будет достигнут какой-то лист. Например, для ситуации, в которой `Patrons = Full` и `WaitEstimate = 0-10` будет получено положительное решение, т.е. следует дождаться освобождения столика.

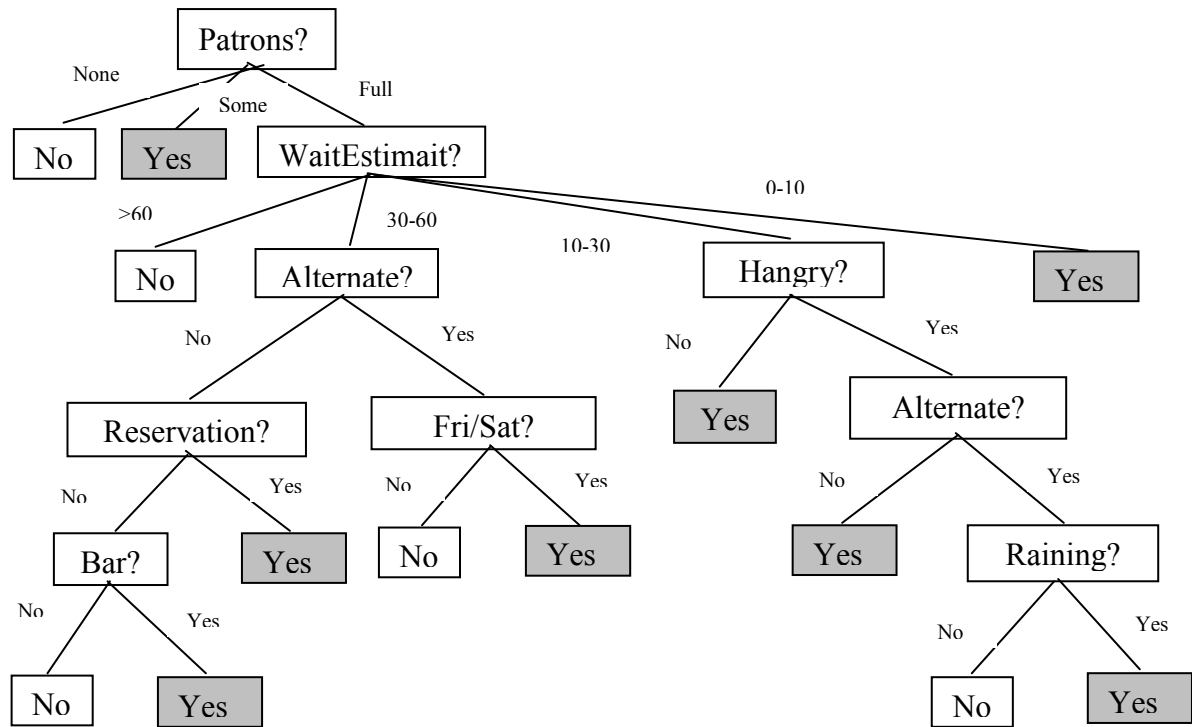


Рисунок 3 – Пример дерева решений