Содержание

ГЛАВА 7

Ассоциативные машины

7.1. Введение

В предыдущих трех главах описывались три различных подхода к обучению с учителем. В главе 4 описывался многослойный персептрон, обучаемый по методу обратного распространения и реализующий одну из форм глобальной оптимизации (во всем пространстве весовых коэффициентов). Сети на основе радиальных базисных функций, описанные в главе 5, благодаря своей структуре обеспечивают локальную оптимизацию. Машины опорных векторов, рассмотренные в главе 6, базируются на теории VC-измерений. В этой главе мы обсудим еще один класс методов, предназначенных для решения задач обучения с учителем. Представленный здесь подход основан на общеизвестном принципе “разделяй и властвуй” (divide and conquer).

В соответствии с этим принципом сложные вычислительные задачи решаются при помощи их разбиения на множество небольших и простых задач с последующим объединением полученных решений. При обучении с учителем вычислительная простота достигается за счет распределения задачи обучения среди множества *экспертов*, которые, в свою очередь, разбивают входное пространство на множество подпространств. Комбинацию таких экспертов и называют *ассоциативной машиной* (committee machine). По сути она интегрирует знания, накопленные экспертами, в общее решение, которое имеет приоритет над каждым решением отдельного эксперта. Идея ассоциативной машины появилась еще в 1965 году в [786]. Предложенная в этой работе сеть состоит из слоя элементарных персептронов, за которым следует второй слой — принятия решения.

Ассоциативные машины являются универсальными аппроксиматорами. Их можно разбить на две основные категории.

1. *Статические структуры* (static structure). В этом классе ассоциативных машин отклики различных предикторов (экспертов) объединяются с помощью некоторого механизма, не учитывающего входной сигнал. Поэтому они и получили название “статические”. Эта категория структур работает на основе следующих методов.

- *Усреднение по ансамблю* (ensemble averaging). Выходной сигнал вычисляется как линейная комбинация выходов отдельных предикторов.
- *Усиление* (boosting), при котором слабый алгоритм обучения превращается в алгоритм, достигающий произвольной заданной точности.

2. *Динамические структуры* (dynamic structure). В этом втором классе ассоциативной машины входной сигнал непосредственно учитывается в механизме объединения выходных сигналов экспертов (благодаря этому свойству данные машины и получили название "динамических"). Можно выделить две различные реализации динамических структур.
   - *Смешение мнений экспертов* (mixture of experts), при котором отклики отдельных экспертов нелинейно объединяются в единую шлюзовую сеть (gating network).
   - *Иерархическое объединение мнений экспертов*, при котором отклики отдельных экспертов нелинейно объединяются с помощью нескольких шлюзовых сетей, организованных в иерархическую структуру.

   При смешении мнений экспертов принцип "разделяй и властвуй" применяется всего один раз, в то время как при иерархическом смешении он применяется неоднократно для каждого слоя иерархии.

Смешение мнений экспертов и иерархическое смешение можно рассматривать как примеры *модульных* (modular) сетей. Формальное определение *модульности* было приведено в [804]. Оно звучит следующим образом.

> *Нейронная сеть является модульной, если выполняемые ею вычисления можно распределить по нескольким подсистемам, которые обрабатывают различные входные сигналы и не пересекаются в своей работе друг с другом. Выходные сигналы этих подсистем объединяются модулем интеграции, выход которого не имеет обратной связи с подсистемами. В частности, модуль интеграции принимает решение о том, как выходные сигналы подсистем объединяются в общий выходной сигнал системы, и определяет, на каких примерах следует обучать конкретные модули.*

Это определение модульности исключает из рассмотрения класс статических ассоциативных машин, так как в последних не существует никаких элементов интегрирования, которые выполняют принятие решений.

Структура главы

Эту главу формально можно разбить на две части. В первой части (разделы 7.2–7.5) рассматривается класс статических структур. В частности, в разделе 7.2 описывается метод усреднения по ансамблю, компьютерное моделирование которого будет выполнено в разделе 7.3. В разделе 7.4 мы рассмотрим технику усиления, а в разделе 7.5 приведем результаты ее компьютерного моделирования.

Во второй части главы (разделы 7.6–7.13) рассматриваются динамические структуры. В частности, в разделе 7.6 представлена структура смешения мнений экспертов, или ME-структура (mixture of experts), как ассоциативная гауссова модель смешения (associative Gaussian mixture model). В разделе 7.7 описывается более общий случай — иерархическое смешение мнений экспертов, или HME-структура (hierarchical mixture of experts). Эта вторая модель близка к стандартным деревьям решений. В разделе 7.8 речь пойдет о том, как стандартное дерево решений можно использовать для решения задачи выбора модели (т.е. количества экспертных и шлюзовых сетей) в HME. В разделе 7.9 будут определены апостериорные вероятности, используемые при описании алгоритмов обучения HME. В разделе 7.10 будут заложены основы для решения задачи оценки параметров и построена функция правдоподобия для модели HME. В разделе 7.11 представлен обзор стратегий обучения, а в разделе 7.12 последует детальное описание так называемого *алгоритма EM*, применению которого для модели HME посвящен раздел 7.13.

Как всегда, глава завершится заключительными выводами и обсуждением.

7.2. Усреднение по ансамблю

На рис. 7.1 показано множество отдельно обучаемых нейронных сетей (экспертов) с общим входным сигналом. Их выходные сигналы некоторым образом комбинируются, формируя общий выход системы y. Для того чтобы упростить выкладки, предположим, что выходы экспертов представляют собой скалярные величины. Представленный здесь подход носит название *метода усреднения по ансамблю* (ensemble averaging method)[1]. Использование этого метода обусловлено двумя основными причинами.

- Если множество экспертов, показанное на рис. 7.1, заменить единой нейронной сетью, получится сеть, содержащая гораздо большее количество настраиваемых параметров. Естественно, время обучения такой сети будет существенно больше времени параллельного обучения множества экспертов.
- Риск избыточного обучения (overfitting) возрастает, если количество настраиваемых параметров существенно больше размера множества данных обучения.

При использовании ассоциативных машин (см. рис. 7.1) предполагается, что обучаемые по отдельности эксперты будут сходиться к разным локальным минимумам поверхности ошибок, в результате чего некоторая комбинация их выходных сигналов приведет к повышению эффективности сети.

[1] Методы усреднения по множеству обсуждаются в [828], в которой собрана довольно большая библиография по этой теме. К другим рекомендуемым работам относятся [424] и [1164].

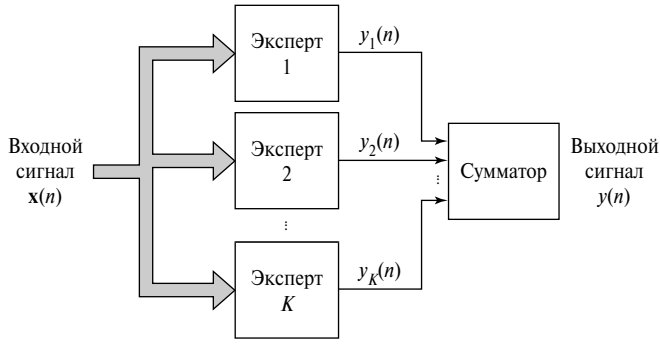

Рис. 7.1. Блочная диаграмма ассоциативной машины, основанной на усреднении по ансамблю

В первую очередь рассмотрим единую нейронную сеть, обучаемую на некотором множестве примеров. Пусть $\mathbf{x}$ — некоторый ранее не встречавшийся входной вектор; d — соответствующий ему желаемый отклик (метка класса или численный отклик). Эти величины представляют собой реализацию случайного вектора $\mathbf{X}$ и случайной переменной D. Пусть $F(\mathbf{x})$ — реализуемая сетью функция отображения входного сигнала в выходной. Тогда в свете дилеммы смещения и дисперсии, описанной в главе 2, среднеквадратическую ошибку между функцией $F(\mathbf{x})$ и условным математическим ожиданием $E(D|\mathbf{X} = \mathbf{x})$ можно разложить на слагаемые смещения и дисперсии

$$E_{\mathbf{D}}[(F(\mathbf{x}) - E[D|\mathbf{X} = \mathbf{x}])^2] = B_{\mathbf{D}}(F(\mathbf{x})) + V_{\mathbf{D}}(F(\mathbf{x})), \tag{7.1}$$

где $B_{\mathbf{D}}(F(\mathbf{x}))$ — квадрат смещения

$$B_{\mathbf{D}}(F(\mathbf{x})) = (E_{\mathbf{D}}[F(\mathbf{x})] - E[D|\mathbf{X} = \mathbf{x}])^2, \tag{7.2}$$

а $V_{\mathbf{D}}(F(\mathbf{x}))$ — дисперсия

$$V_{\mathbf{D}}(F(\mathbf{x})) = E_{\mathbf{D}}[(F(\mathbf{x}) - E_{\mathbf{D}}[F(\mathbf{x})])^2]. \tag{7.3}$$

Математическое ожидание $E_{\mathbf{D}}$ вычисляется в пространстве $\mathbf{D}$, которое определяется как *пространство, охватывающее распределение всех обучающих множеств (т.е. входных сигналов и целевых выходов) и распределение всех начальных условий*.

Существуют различные способы отдельного обучения различных экспертных сетей, а также различные методы объединения их выходных сигналов. В данной главе будет рассмотрена ситуация, в которой экспертные сети имеют одинаковую конфигурацию, но начинают обучение из различных исходных состояний. Для объединения их входных сигналов будет использоваться простой *блок усреднения по ансамблю*[2].

[2] Использование усреднения по ансамблю при построении ассоциативной машины на множестве различных начальных условий ранее предлагалось многими практиками нейронных сетей. Однако статистический анализ, представленный в [770], и процедура обучения ассоциативных машин, определенная для метода усред-

Пусть $\mathbf{\Phi}$ — *пространство всех исходных состояний*; $F_I(\mathbf{x})$ — среднее всех функций отображения входного сигнала в выходной в экспертных сетях на множестве исходных состояний. По аналогии с формулой (7.1) можно записать следующее:

$$E_{\mathbf{\Phi}}[(F_I(\mathbf{X}) - E[D|\mathbf{X} = \mathbf{x}])^2] = B_{\mathbf{\Phi}}(F(\mathbf{x})) + V_{\mathbf{\Phi}}(F(\mathbf{x})), \tag{7.4}$$

где $B_{\mathbf{\Phi}}(F(\mathbf{x}))$ — квадрат смещения, определенного в пространстве $\mathbf{\Phi}$:

$$B_{\mathbf{\Phi}}(F(\mathbf{x})) = (E_{\mathbf{\Phi}}[F_I(\mathbf{x})] - E[D|\mathbf{X} = \mathbf{x}])^2, \tag{7.5}$$

а $V_{\mathbf{\Phi}}(F(\mathbf{x}))$ — соответствующая дисперсия:

$$V_{\mathbf{\Phi}}(F(\mathbf{x})) = E_{\mathbf{\Phi}}[(F_I(\mathbf{x}) - E_{\mathbf{\Phi}}[F(\mathbf{x})])^2]. \tag{7.6}$$

Математическое ожидание $E_{\mathbf{\Phi}}$ вычисляется по всему пространству $\mathbf{\Phi}$.

Из определения пространства $\mathbf{D}$ видно, что оно представляет собой произведение пространства исходных условий $\mathbf{\Phi}$ на *оставшееся пространство* (remnant space) $\mathbf{D}'$. Следовательно, по аналогии с (7.1) можно записать:

$$E_{\mathbf{D}'}[(F_I(\mathbf{x}) - E[D|\mathbf{X} = \mathbf{x}])^2] = B_{\mathbf{D}'}(F_I(\mathbf{x})) + V_{\mathbf{D}'}(F_I(\mathbf{x})), \tag{7.7}$$

где $B_{\mathbf{D}'}(F(\mathbf{x}))$ — квадрат смещения, определенного в пространстве $\mathbf{\Phi}'$:

$$B_{\mathbf{D}'}(F_I(\mathbf{x})) = (E_{\mathbf{D}'}[F_I(\mathbf{x})] - E[D|\mathbf{X} = \mathbf{x}])^2, \tag{7.8}$$

а $V_{\mathbf{D}'}(F(\mathbf{x}))$ — соответствующая дисперсия:

$$V_{\mathbf{D}'}(F_I(\mathbf{x})) = E_{\mathbf{D}'}[(F_I(\mathbf{x}) - E_{\mathbf{D}'}[F_I(\mathbf{x})])^2]. \tag{7.9}$$

Из определений пространств $\mathbf{D}$, $\mathbf{\Phi}$ и $\mathbf{D}'$ видно, что

$$E_{\mathbf{D}'}[F_I(\mathbf{x})] = E_{\mathbf{D}}[F_I(\mathbf{x})]. \tag{7.10}$$

нения по множеству начальных условий, явились первыми исследованиями такого рода. Представленные в этой работе экспериментальные результаты подтвердили значительное уменьшение дисперсии при усреднении по пространству начальных условий.

Согласно [770], в ассоциативных машинах, использующих усреднение по пространству начальных состояний, *не рекомендуется* применять популярные модификации алгоритма обучения, типа уменьшения весов или раннего останова.

Отсюда следует, что выражение (7.8) можно переписать в эквивалентной форме:

$$B_{\mathbf{D}'}(F_I(\mathbf{x})) = (E_{\mathbf{D}}[F(\mathbf{x})] - E[D|\mathbf{X} = \mathbf{x}])^2 = B_{\mathbf{D}}(F(\mathbf{x})). \tag{7.11}$$

Далее рассмотрим дисперсию $V_{\mathbf{D}'}(F(\mathbf{x}))$ в выражении (7.9). Так как дисперсия случайной переменной равна среднеквадратическому значению этой переменной за вычетом квадрата смещения, то можно записать:

$$V_{\mathbf{D}'}(F_I(\mathbf{x})) = E_{\mathbf{D}'}[(F_I(\mathbf{x}))^2] - (E_{\mathbf{D}'}[F_I(\mathbf{x})])^2 = E_{\mathbf{D}'}[(F_I(\mathbf{x}))^2] - (E_{\mathbf{D}}[F_I(\mathbf{x})])^2. \tag{7.12}$$

В последнем равенстве используется соотношение (7.10). Аналогично, выражение (7.3) можно переписать в следующей эквивалентной форме:

$$V_{\mathbf{D}}(F_I(\mathbf{x})) = E_{\mathbf{D}}[(F(\mathbf{x}))^2] - (E_{\mathbf{D}}[F(\mathbf{x})])^2. \tag{7.13}$$

Заметим, что среднеквадратическое значение функции $F(\mathbf{x})$ во всем пространстве $\mathbf{D}$ должно быть не меньше значения среднеквадратической функции $F_I(\mathbf{x})$ в пространстве дополнения $\mathbf{D}'$, т.е.

$$E_{\mathbf{D}}[F(\mathbf{x})^2] \geq E_{\mathbf{D}'}[(F_I(\mathbf{x}))^2].$$

В свете этого неравенства сравнение (7.12) и (7.13) приводит к следующему заключению:

$$V_{\mathbf{D}'}(F_I(\mathbf{x})) \leq V_{\mathbf{D}}(F(\mathbf{x})). \tag{7.14}$$

Таким образом, из выражений (7.11) и (7.14) можно сделать следующие два вывода.

1. Смещение усредненной по ансамблю функции $F_I(\mathbf{x})$ для ассоциативной машины, показанной на рис. 7.1, имеет то же значение, что и для функции $F(\mathbf{x})$ отдельной нейронной сети.
2. Дисперсия усредненной по ансамблю функции $F_I(\mathbf{x})$ не меньше дисперсии отдельных функций $F(\mathbf{x})$.

Эти теоретические рассуждения определяют стратегию обучения, которая приводит к уменьшению общей ошибки ассоциативной машины за счет *варьирования начальных состояний* [770]: отдельные "эксперты" машины целенаправленно *обучаются с избытком* (overtrained). Этому есть следующее объяснение: поскольку мы имеем дело с отдельными экспертами, смещение уменьшается за счет дисперсии. Далее дисперсия уменьшается путем усреднения параметров экспертов ансамбля по начальным условиям при неизменном смещении.

7.3. Компьютерный эксперимент 1

В этом компьютерном эксперименте, посвященном исследованию метода *усреднения по ансамблю*, снова рассмотрим задачу бинарной классификации, с которой мы уже сталкивались в предыдущих трех главах. Это задача классификации двух классов, определяемых двумерными пересекающимися гауссовыми распределениями. Эти два распределения имеют различные средние значения и дисперсии. Класс $\mathbf{C}_1$ имеет такие статистические характеристики:

$$\boldsymbol{\mu}_1 = [0, 0]^T,$$
$$\sigma_1^2 = 1,$$

а класс $\mathbf{C}_2$ — следующие:

$$\boldsymbol{\mu}_2 = [2, 0]^T,$$
$$\sigma_2^2 = 4.$$

Графики этих двух распределений представлены на рис. 4.13.

Предполагается, что эти два класса равновероятны. Стоимость ошибки классификации предполагается равной в обоих классах, а стоимость корректной классификации предполагается равной нулю. При таких условиях (оптимальный) классификатор Байеса обеспечивает вероятность корректной классификации, составляющую p_c =81,51%. Обоснование этого результата приводится в главе 4.

В компьютерном эксперименте, описанном в главе 4, для персептрона с двумя скрытыми элементами, обучаемого с помощью алгоритма обратного распространения, удалось обеспечить вероятность корректной классификации порядка 80%. В настоящем разделе для решения задачи будем использовать ассоциативную машину, состоящую из десяти экспертов, при этом каждый из экспертов представляет собой многослойный персептрон с двумя скрытыми нейронами.

Все эксперты обучаются отдельно с помощью алгоритма обратного распространения со следующими параметрами.

Параметр скорости обучения η =0,1.

Константа фактора момента α =0,5.

Обучающее множество состоит из 500 образов. Все эксперты обучаются на одном и том же множестве примеров, но имеют различные исходные состояния. В частности, исходные значения синаптических весов и порогов выбираются случайным образом с помощью генератора случайных чисел с равномерным распределением в диапазоне $[-1; 1]$.

ТАБЛИЦА 7.1. Эффективность классификации отдельных экспертов в ассоциативной машине

Эксперт	*Процент корректной классификации*
Сеть 1	80,65
Сеть 2	76,91
Сеть 3	80,06
Сеть 4	80,47
Сеть 5	80,44
Сеть 6	76,89
Сеть 7	80,55
Сеть 8	80,47
Сеть 9	76,91
Сеть 10	80,38

В табл. 7.1 представлены результаты классификации отдельных экспертов, обучаемых на данном множестве образов. Вероятность корректной классификации была получена путем вычисления среднего арифметического по множеству результатов в табл. 7.1 и составила $p_{c,\mathrm{av}}$ =79,37%. С другой стороны, используя метод *усреднения по ансамблю*, т.е. просто суммируя результаты всех 10 экспертов и только затем вычисляя вероятность корректной классификации, получим результат $p_{c,\mathrm{ens}} = 80,27\%$. Налицо улучшение результата на 0,9%. Более высокое значение $p_{c,\mathrm{ens}}$ по сравнению с $p_{c,\mathrm{av}}$ наблюдалось во всех попытках данного эксперимента. Результаты классификации каждый раз вычислялись на тестовом множестве из 32000 примеров.

Анализируя результаты эксперимента, можно утверждать следующее. Эффективность классификации улучшается за счет переобучения отдельных персептронов (экспертов) и суммирования их выходных сигналов в единый выходной сигнал ассоциативной машины, на основании которого уже и принимается решение.

7.4. Метод усиления

Как уже говорилось во введении, метод усиления является еще одним способом реализации класса “статических” ассоциативных машин. В ассоциативных машинах, основанных на усреднении по ансамблю, все эксперты обучаются на одном и том же множестве данных. Сети отличаются друг от друга только выбором исходного состояния. В противоположность этому сети-эксперты, работающие на основе метода усиления, обучаются на примерах, принадлежащих совершенно различным распределениям. Это самый общий метод из тех, которые можно использовать для улучшения производительности *любого* алгоритма обучения.

Метод усиления[3] (boosting) может быть реализован тремя способами.

1. *Усиление за счет фильтрации* (boosting by filtering). Этот подход предполагает отбор (фильтрацию) примеров обучения различными версиями слабого алгоритма обучения. При этом предполагается доступность большого (в идеале — бесконечного) множества примеров. Во время обучения примеры могут быть отбракованы или сохранены. Преимуществом этого подхода является то, что по сравнению с двумя остальными он не предъявляет больших требований к памяти.
2. *Усиление за счет формирования подвыборок* (boosting by subsampling). Этот подход предполагает наличие множества примеров обучения фиксированного размера. Подвыборки составляются во время обучения в соответствии с заданным распределением вероятности. Ошибка вычисляется относительно фиксированного множества примеров обучения.
3. *Усиление путем перевзвешивания* (boosting by reweighting). Третий подход связан с обработкой фиксированного множества примеров. При этом предполагается, что слабый алгоритм обучения может получать "взвешенные" примеры. Ошибка вычисляется относительно взвешенных примеров.

В этом разделе описываются два алгоритма усиления. Первый из них, согласно [940], относится к первому вышеописанному подходу. Второй алгоритм, называемый AdaBoost [319], [320], относится ко второму подходу.

Усиление за счет фильтрации

Исходная идея усиления, описанная в [940], берет свое начало в *независимой от распределения* (distribution-free) или *статистически аппроксимативно корректной* (probably approximately correct — PAC) модели обучения. Из обсуждения модели PAC в главе 2 ясно, что *понятие* или *концепт* (concept) можно рассматривать как булеву функцию в предметной области экземпляров, которая включает коды всех интересующих нас объектов. При обучении PAC-машина пытается идентифицировать неизвестный двоичный концепт на основе случайно выбранных его экземпляров. Более строго, целью обучаемой машины является поиск гипотезы или правила прогнозирования с вероятностью ошибки, не превышающей некоторой небольшой наперед заданной величины ε, причем это условие должно выполняться для всех входных распределений. Именно по этой причине обучение PAC называют также *сильной моделью обучения* (strong learning model). Так как примеры имеют случайную природу, обучаемая машина, вероятнее всего, не сможет составить представление о неизвестном понятии, если множество примеров будет не представительным. Поэтому модель обучения должна

[3] Основными работами по теории усиления и связанными с ней экспериментальными приложениями являются следующие: [940], [267], [266], [317], [153], [319], [320], [318], [939] и [941]. Они перечислены примерно в хронологическом порядке Лучшими ссылками по трем основным подходам к усилению являются [940] (фильтрация), [319] (подвыборка (resampling)) и [317] (перевзвешивание).

находить хорошую аппроксимацию неизвестного понятия с вероятностью $(1-\delta)$, где δ — некоторое малое положительное число.

В вариации PAC-обучения, называемой *слабой моделью обучения* (weak learning model), требования к поиску неизвестного понятия сильно ослаблены. Обучаемая машина должна построить гипотезу с вероятностью ошибки, несколько меньшей значения 1/2. При случайном "угадывании" двоичной разметки для каждого примера гипотеза с равной вероятностью может оказаться как правильной, так и неверной. Значит, в этом случае вероятность ошибки составляет 1/2. Отсюда следует, что для слабой модели обучения достаточно достичь уровня эффективности, несколько превосходящего абсолютно случайный выбор. Понятие слабой обучаемости было введено в [550], где была сформулирована *задача усиления гипотезы* (hypothesis boosting problem), которая сводится к следующему вопросу.

Эквивалентны ли понятия сильного и слабого обучения?

Другими словами, является ли любой слабо обучаемый класс понятий также и сильно обучаемым? Положительный ответ на этот вопрос, несколько удивительный на первый взгляд, дан в [940], где представлено конструктивное доказательство эквивалентности слабого и сильного обучения. В частности, в [940] был предложен алгоритм, преобразующий слабую модель обучения в сильную. Это достигается благодаря изменению распределения примеров обучения таким образом, чтобы сильная модель строилась "вокруг" слабой.

При использовании усиления за счет фильтрации ассоциативная машина состоит из трех экспертов или подгипотез. Алгоритм, используемый для такого обучения, называют алгоритмом *усиления* (boosting algorithm). При этом три эксперта произвольно маркируются как "первый", "второй" и "третий". Они обучаются по отдельности следующим образом.

1. Первый эксперт обучается на множестве, состоящем из N_1 примеров.
2. Обученный первый эксперт используется для *фильтрации* (filter) второго множества примеров следующим образом.
   - Случайный выбор моделируется подбрасыванием монетки.
   - Если выпадает *решка* (head), новый пример "пропускается" через первого эксперта и корректно классифицированные примеры отклоняются до тех пор, пока не возникнет ошибка классификации. Пример, приведший к ошибке классификации, добавляется в множество примеров для обучения второго эксперта.
   - Если выпадает *орел* (tail), производятся действия, прямо противоположные вышеописанным, т.е. примеры "пропускаются" через первого эксперта и отклоняются до тех пор, пока очередной пример не будет классифицирован правильно. Этот корректно классифицированный пример добавляется в множество примеров, подготавливаемых для обучения второго эксперта.

- Этот процесс продолжается до тех пор, пока все множество из N_1 примеров не будет отфильтровано первым экспертом. Отфильтрованное таким образом множество примеров подается для обучения второго эксперта.

Процедура подбрасывания монетки гарантирует, что при тестировании первого эксперта на втором наборе примеров ошибка классификации составит 1/2. Другими словами, второе множество из N_1 примеров, доступное для обучения второго эксперта, имеет распределение, полностью отличное от распределения первого множества из N_1 примеров, использованных ранее для обучения первого эксперта. Таким образом, второй эксперт вынужден учиться на распределении, отличающемся от использованного для обучения первого эксперта.

3. После обучения второго эксперта множество примеров обучения для третьего эксперта формируется следующим образом.
   - Новый пример "пропускается" через первого и второго экспертов. Если решения обоих экспертов совпадают, пример отклоняется; если они расходятся в своих мнениях, данный пример включается в множество примеров обучения третьего эксперта.
   - Этот процесс продолжается до тех пор, пока не будет отфильтровано все множество из N_1 примеров. Полученное множество примеров затем используется для обучения третьего эксперта.

   После окончания обучения третьего эксперта на отфильтрованном множестве примеров процесс обучения всей ассоциативной машины считается завершенным.

Описанная выше процедура фильтрации графически представлена на рис. 7.2.

Пусть N_2 — количество примеров, отфильтрованных первым экспертом для получения обучающего множества второго эксперта, содержащего N_1 элементов. Заметим, что число N_1 фиксировано, а число N_2 зависит от ошибки обобщения первого эксперта. Пусть N_3 — количество примеров, обработанных в процессе совместной фильтрации первым и вторым экспертами для формирования обучающего множества третьего эксперта из N_1 элементов. Так как это же число (N_1) ранее использовалось для обучения первого эксперта, общий объем данных, необходимый для обучения ассоциативной машины, составляет $N_4 = N_1 + N_2 + N_3$. Однако при этом вычислительная стоимость обучения зависит от 3*N_1 примеров, так как N_1 — это количество примеров, действительно используемых для обучения всех трех экспертов. Таким образом, можно утверждать, что описанный здесь алгоритм усиления является действительно "интеллектуальным" (smart): для работы ассоциативной машины требуется большой объем примеров, но на самом деле для реального обучения будет использоваться только небольшое подмножество этих данных.

Рис. 7.2. Графически представленный процесс усиления за счет фильтрации

Еще одним вопросом, на который хотелось бы обратить внимание, является следующий. Операция фильтрации, выполняемая первым экспертом, и операция совместной фильтрации, совместно выполняемая первым и вторым экспертами, заставляют второго и третьего экспертов сосредоточиться на "тяжелых для усвоения" частях распределения.

В теоретическом выводе алгоритма усиления, впервые представленном в [940], для оценки производительности ассоциативной машины на не встречавшихся ранее примерах использовалось простое голосование. А именно, на вход ассоциативной машины подавался тестовый пример. Если решения первого и второго экспертов совпадали, для маркировки использовался именно этот класс. В противном случае использовалась оценка третьего эксперта. Однако экспериментально [266], [267] было определено, что *сложение* соответствующих выходов трех экспертов приводит к лучшей производительности, нежели голосование. Например, в задаче оптического распознавания символов (optical character recognition — OCR) операция сложения выполнялась сложением выходов "цифра 0" всех трех экспертов; аналогично — для всех остальных девяти цифр.

Предположим, что все три эксперта (т.е. подгипотезы) имеют уровень ошибок $\varepsilon < 1/2$ по отношению к распределениям, на которых происходило их обучение. Это значит, что все три модели обучения являются слабыми. В [940] было доказано, что общий уровень ошибок ассоциативной машины в данном случае ограничен следующей величиной:

$$g(\varepsilon) = 3\varepsilon^2 - 2\varepsilon^3. \tag{7.15}$$

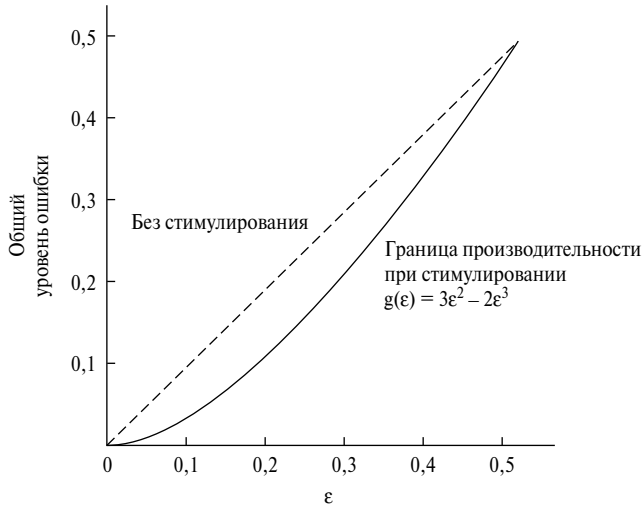

Рис. 7.3. График зависимости (7.15) для усиления за счет фильтрации

Зависимость $g(\varepsilon)$ от ε показана на рис. 7.3. На рисунке видно, что это ограничение существенно меньше исходного уровня ошибок ε. Рекурсивно применяя алгоритм усиления, уровень ошибок можно сделать сколь угодно малым. Другими словами, модель слабого обучения, производительность которой мало отличается от случайного выбора, при помощи усиления можно преобразовать в сильную модель обучения. Именно в этом смысле можно с полным правом утверждать, что слабая и сильная модели обучения на самом деле эквивалентны.

Алгоритм адаптивного усиления AdaBoost

Практическое ограничение усиления за счет фильтрации состоит в том, что для него часто требуется довольно большое множество примеров. Это ограничение можно обойти, используя другой алгоритм усиления, который называется *AdaBoost* [319], [320] и принадлежит к категории усиления с использованием подвыборки. Контур подвыборки AdaBoost является обычным контуром пакетного обучения. И, что более важно, — он допускает повторное использование данных обучения.

Как и в случае использования алгоритма усиления за счет фильтрации, AdaBoost позволяет работать со слабыми моделями обучения. Целью этого алгоритма является поиск окончательной функции отображения или гипотезы, которая будет иметь низкий уровень ошибок на данном распределении **D** маркированных примеров обучения. От других алгоритмов усиления AdaBoost отличается следующим.

- AdaBoost *адаптивно* настраивается на ошибки слабых гипотез, возвращаемых слабыми моделями обучения. Благодаря именно этому свойству алгоритм и получил свое название.

- Ограничение производительности алгоритма AdaBoost зависит от производительности слабой модели обучения только на тех распределениях, которые фактически формируются в процессе обучения.

Алгоритм AdaBoost работает следующим образом. На итерации n алгоритм *усиления* реализует слабую модель обучения с распределением $\mathbf{D}_n$ множества примеров обучения $\mathbf{T}$. Слабая модель обучения вычисляет гипотезу $\mathbf{F}_n$: $\mathbf{X} \rightarrow Y$, которая корректно классифицирует часть примеров обучения. Ошибка измеряется относительно распределения $\mathbf{D}_n$. Этот процесс продолжается T итераций, в результате чего машина *усиления* объединяет гипотезы $\mathbf{F}_1$, $\mathbf{F}_2$,. . . ,$\mathbf{F}_T$ в единую заключительную гипотезу $\mathbf{F}_{\text{fin}}$.

Для того чтобы вычислить распределение $\mathbf{D}_n$ на итерации n и заключительную гипотезу $\mathbf{F}_{\text{fin}}$, используется простая процедура, описанная вкратце в табл. 7.2. Исходное распределение является равномерным на множестве примеров обучения $\mathbf{T}$, т.е.

$$\mathbf{D}_1(i) = \frac{1}{n} \text{ для всех } i.$$

Для данного распределения $\mathbf{D}_n$ и слабой гипотезы $\mathbf{F}_n$ на итерации n следующее распределение $\mathbf{D}_{n+1}$ вычисляется умножением веса примера i на некоторое число $\beta_n \in [0, 1)$, если гипотеза $\mathbf{F}_n$ корректно классифицирует входной вектор $\mathbf{x}_i$. В противном случае вес остается неизменным. В итоге все веса заново нормализуются делением на константу нормализации Z_n. В результате "легкие" примеры множества $\mathbf{T}$, корректно классифицированные предыдущей слабой гипотезой, будут иметь более низкий вес, в то время как "трудные" примеры, для которых частота ошибок классификации была высокой, будут иметь больший вес. Таким образом, алгоритм AdaBoost концентрирует основной объем весов на тех примерах, классифицировать которые оказалось сложнее всего.

Окончательная гипотеза $\mathbf{F}_{\text{fin}}$ вычисляется голосованием (т.е. взвешенным линейным порогом) из множества гипотез $\mathbf{F}_1$, $\mathbf{F}_2$,. . . ,$\mathbf{F}_T$. Это значит, что для данного входного вектора $\mathbf{x}$ окончательная гипотеза дает на выходе метку d, которая максимизирует сумму весов слабых гипотез, которые определили эту метку. Вес гипотезы $\mathbf{F}_n$ определяется как $\log(1/\beta_n)$, т.е. больший вес назначается гипотезе с меньшей ошибкой.

Важное теоретическое свойство алгоритма AdaBoost было сформулировано в следующей теореме [319].

Пусть при вызове алгоритма AdaBoost слабая модель обучения сгенерировала гипотезы с ошибками $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_T$, *где ошибка* ε_n *на итерации* n *алгоритма AdaBoost определяется как*

$$\varepsilon_n = \sum_{i:\mathbf{F}_n(\mathbf{x}_i) \neq d_i} \mathbf{D}_n(i).$$

ТАБЛИЦА 7.2. Краткое описание алгоритма AdaBoost

Дано:	Множество примеров обучения $\{(x_i, d_i)\}_{i=1}^N$
	Распределение $\mathbf{D}$ для N маркированных примеров
	Слабая модель обучения
	Целое число T, задающее количество итераций алгоритма
Инициализация:	$\mathbf{D}_1(i) = 1/N$ для всех i
Вычисления:	Для $n = 1, 2, \ldots, T$ выполняются следующие действия.

1. Вызывается слабая модель обучения, реализуемая на распределении $\mathbf{D}_n$
2. Формируется гипотеза $\mathbf{F}_n$: $\mathbf{X} \to Y$
3. Вычисляется ошибка гипотезы $\mathbf{F}_n$
$$\varepsilon_n = \sum_{i:\mathbf{F}_n(\mathbf{x}_i) \neq d_i} \mathbf{D}_n(i)$$
4. Вычисляется $\beta_n = (\varepsilon_n)/(1 - \varepsilon_n)$
5. Изменяется распределение $\mathbf{D}_n$:
$$\mathbf{D}_{n+1}(i) = \frac{\mathbf{D}_n(i)}{Z_n} \times \begin{cases} \beta_n, & \text{если } \mathbf{F}_n(\mathbf{x}_i) = d_i, \\ 1, & \text{в противном случае,} \end{cases}$$
где Z_n — константа нормализации (выбираемая таким образом, чтобы $\mathbf{D}_{n+1}$ было распределением вероятности)

Выход: Окончательная гипотеза вычисляется следующим образом:
$$\mathbf{F}_n(\mathbf{x}) = \arg\max_{d \in \mathbf{D}} \sum_{n:\mathbf{F}_n(\mathbf{x})=d} \log \frac{1}{\beta_n}$$

Пусть также $\varepsilon_n \leq 1/2$ *и* $\gamma_n = 1/2 - \varepsilon_n$. *Тогда ошибка окончательной гипотезы ограничена сверху следующей величиной:*

$$\frac{1}{N} |\{i : \mathbf{F}_{\text{fin}}(\boldsymbol{x}_i) \neq d_i\}| \leq \prod_{n=1}^{T} \sqrt{1 - 4\gamma_n^2} \leq \exp\left(-2 \sum_{i=1}^{T} \gamma_n^2\right). \tag{7.16}$$

Согласно этой теореме, если ошибка слабой гипотезы, построенной слабой моделью обучения, лишь немного меньше величины 1/2, то ошибка обучения окончательной гипотезы $\mathbf{F}_{\text{fin}}$ экспоненциально стремится к нулю. Однако это совсем не значит, что ошибка обобщения на тестовых примерах тоже будет мала. Эксперименты, представленные в [319], показали следующее. Во-первых, теоретическая граница ошибки обучения часто оказывается слабой. Во-вторых, ошибка обобщения зачастую существенно лучше, нежели это следует из теоретических выкладок.

В табл. 7.2 представлено описание алгоритма AdaBoost для задачи двоичной классификации.

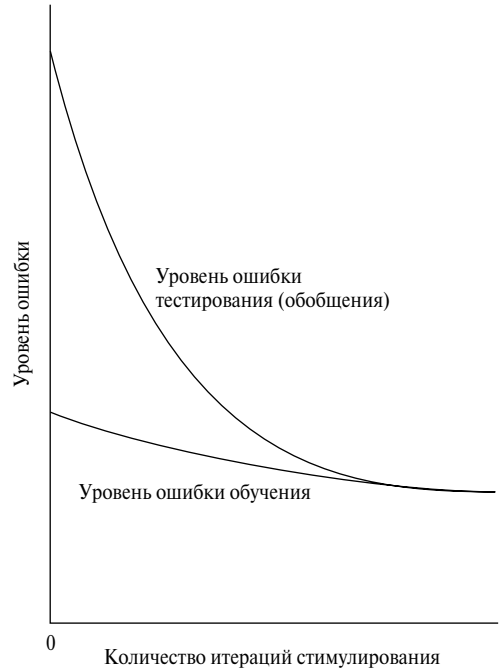

Рис. 7.4. Ошибка алгоритма AdaBoost

Если количество возможных классов (меток) $M>2$, задача усиления становится более интересной ввиду того, что вероятность корректности случайного решения равна $1/M$, что существенно меньше чем 1/2. Для того чтобы алгоритм усиления в такой ситуации мог использовать любую гипотезу, даже слабо отличающуюся от случайного решения, следует несколько модифицировать алгоритм и определение "слабого обучения". Подобные модификации были описаны в [318] и [939].

Изменение ошибки

Эксперименты с алгоритмом AdaBoost, описанные в [153], наглядно показали, что ошибки обучения и тестирования, представленные как функции от количества итераций усиления, имеют следующую зависимость: ошибки тестирования продолжают уменьшаться после того, как ошибки обучения сократились практически до нуля (рис. 7.4). Аналогичный результат был ранее описан в работе, посвященной усилению за счет фильтрации [266].

Эффект, показанный на рис. 7.4, может показаться очень неожиданным в свете того, что нам известно об эффективности обобщения в обычных нейронных сетях. Как известно из главы 4, для многослойного персептрона, обучаемого с помощью алгоритма обратного распространения, ошибка тестирования уменьшается, достигает своего минимума, а затем начинает возрастать в связи с *избыточным обучением*

(overfitting) (см. рис. 4.20). Поведение, показанное на рис. 7.4, в корне отличается тем, что по мере усложнения сети в процессе обучения ошибка обобщения все равно продолжает уменьшаться. Такое поведение противоречит принципу *бритвы Оккама* (Occam's razor), утверждающему, что для обеспечения наилучшего обобщения обучаемая машина должна быть настолько простой, насколько это возможно.

В [941] дано объяснение этому феномену применительно к алгоритму AdaBoost. Ключевой идеей анализа, представленного в работе, является то, что при оценке ошибки обобщения, производимой машиной усиления, следует учитывать не только ошибку обучения, но и *достоверность* (confidence) классификации. Проведенный анализ обнаружил зависимость между усилением и машинами опорных векторов (см. предыдущую главу). В частности, *граница* (margin) классификации определяется как разность между весом, назначенным правильной метке, и максимальным весом, назначенным некоторой некорректной метке. Из этого определения легко увидеть, что граница является числом из диапазона $[-1, +1]$ и образ корректно классифицируется тогда и только тогда, когда его граница имеет положительное значение. Таким образом, в [941] было показано, что явление, наблюдаемое на рис. 7.4, на самом деле связано с распределением границ обучающих примеров по отношению к ошибкам классификации, полученным в результате голосования. Здесь снова можно подчеркнуть, что анализ, проведенный в вышеупомянутой работе, относится только к алгоритму AdaBoost и не применим ни к какому другому алгоритму усиления.

7.5. Компьютерный эксперимент 2

В этом эксперименте мы исследуем алгоритм усиления за счет фильтрации, используемый для решения крайне сложной задачи классификации. Данная задача классификации является двумерной и имеет невыпуклые области решений (рис. 7.5). Первый класс точек лежит в области, помеченной на рисунке $\mathbf{C}_1$, в то время как второй класс — в области $\mathbf{C}_2$. Требуется построить ассоциативную машину, определяющую принадлежность примеров одному из этих двух классов.

Ассоциативная машина, используемая для решения поставленной задачи, состоит из трех экспертов. Каждый из экспертов является многослойным персептроном типа 2-5-2 (два входных узла, пять скрытых и два выходных нейрона). Для обучения используется алгоритм обратного распространения. На рис. 7.6 показан график распределения данных, используемых для обучения всех трех экспертов. На рис. 7.6, *а* показаны данные, используемые для обучения первого эксперта. Данные на рис. 7.6, *б* были отфильтрованы первым экспертом после завершения обучения. Они будут использоваться для обучения второго эксперта. Данные на рис. 7.6, *в* были отфильтрованы в процессе совместной работы первого и второго экспертов. Они будут использоваться для обучения третьего эксперта. Размер множества обучения каждого из экспертов состоял из $N_1 = 1000$ примеров. Из этих трех рисунков видно следующее.

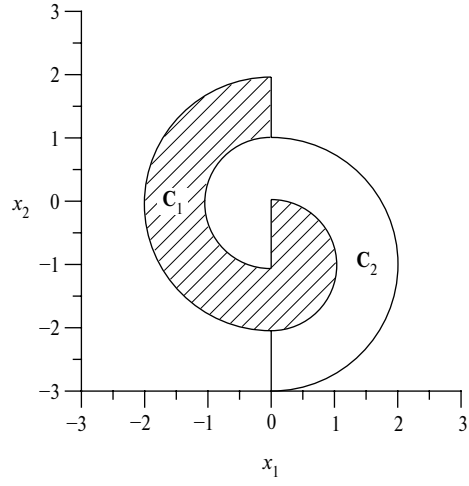

Рис. 7.5. Конфигурация множеств, использованных для моделирования усиления

- Данные, использованные для обучения первого эксперта, распределены равномерно (см. рис. 7.6, *а*).
- Данные, использованные для обучения второго эксперта, более плотно сконцентрированы в областях, помеченных буквами А и Б, т.е. в тех областях, где первый эксперт испытывал сложности при классификации. Количество точек данных в этих двух областях равно количеству правильно классифицированных точек.
- Данные, используемые для обучения третьего эксперта, лежат в области, где первый и второй эксперты испытывали особые сложности при классификации.

На рис. 7.7, *а–в* показаны границы решений, сформированные экспертами 1, 2 и 3 соответственно. На рис. 7.7, *г* показана общая граница решений, сформированная совместными усилиями всех трех экспертов (в данном случае производилось обычное суммирование выходных сигналов отдельных экспертов). Обратите внимание, что различие областей решений первого и второго экспертов (см. рис. 7.7, *а* и *б*) формирует множество точек данных, используемых для обучения третьего эксперта (см. рис. 7.6, *в*).

Вероятность корректной классификации трех экспертов на тестовых данных составила:

1 эксперт: 75,15%
2 эксперт: 71,44%
3 эксперт: 68,90%

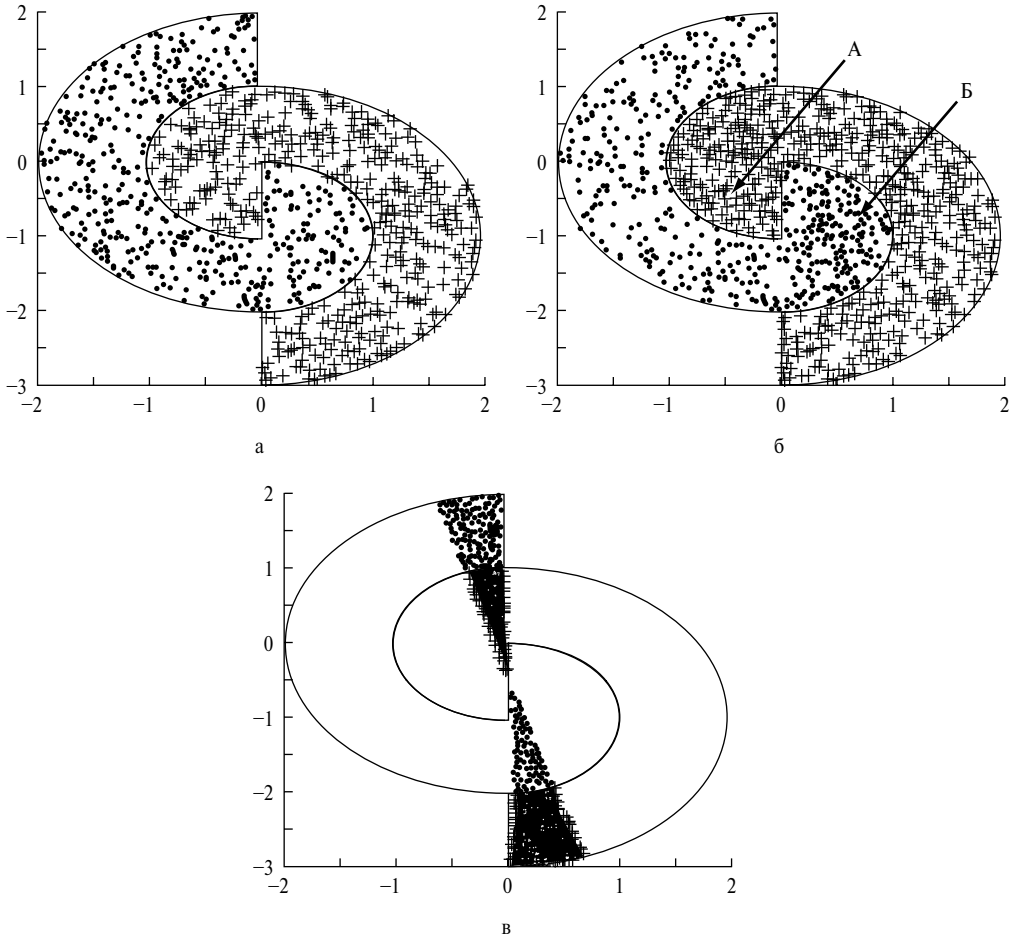

Рис. 7.6. Графики распределения множеств обучения экспертов в компьютерном эксперименте по усилению: эксперт 1 (а); эксперт 2 (б); эксперт 3 (в)

Общая вероятность корректной классификации всей ассоциативной машины составила 91,72%. При вычислении этого показателя использовалось множество тестовых данных, состоящее из 32000 точек. Общая граница решений, сформированная алгоритмом *усиления* с тремя экспертами, показана на рис. 7.7, *г*. Этот рисунок является еще одним доказательством хорошей эффективности классификации.

7.6. Ассоциативная гауссова модель смешения

Вторая часть настоящей главы, которая начинается с этого раздела, будет посвящена изучению второго класса ассоциативных машин — динамических структур. Исполь-

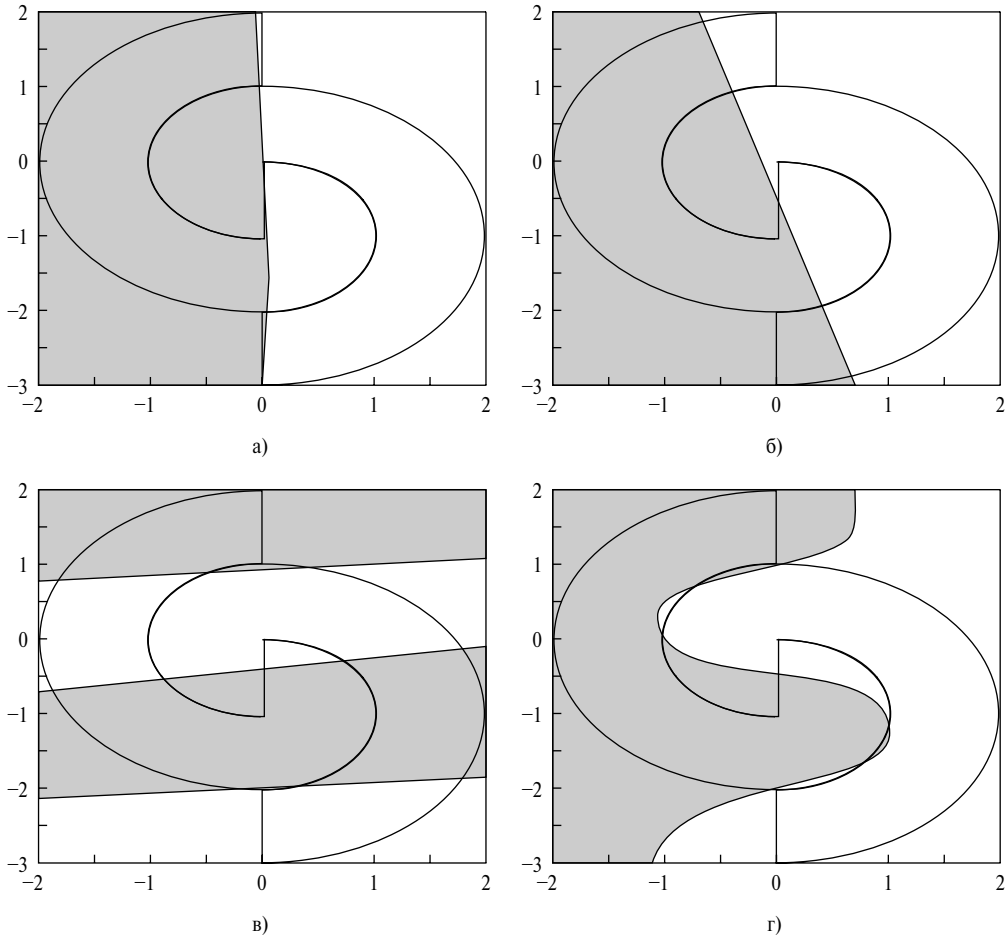

Рис. 7.7. Границы решений, сформированные различными экспертами в эксперименте по усилению: эксперт 1 (а); эксперт 2 (б); эксперт 3 (в); ассоциативная машина в целом (г)

зуемый здесь термин “динамические” подразумевает то, что объединение знаний, накопленных экспертами, происходит при участии самого входного сигнала.

Для того чтобы начать изучение данного вопроса, рассмотрим модульную нейронную сеть, в которой процесс обучения происходит при неявном объединении самоорганизующейся формы обучения и формы обучения с учителем. Эксперты технически обеспечивают обучение с учителем, поскольку их отдельные выходы объединяются для получения желаемого отклика. Однако сами эксперты осуществляют самоорганизующееся обучение. Это значит, что они самоорганизуются с целью нахождения оптимального разбиения входного пространства, причем каждый из них в своем подпространстве имеет наилучшую производительность, а вся группа обеспечивает хорошую модель всего входного пространства.

Описанная схема обучения отличается от схем, рассмотренных в предыдущих трех главах, поскольку здесь для генерации данных обучения предполагается использование специфической модели.

Вероятностная порождающая модель

Для того чтобы понять основную идею, рассмотрим задачу *регрессии* (regression), в которой регрессор $\mathbf{x}$ порождает отклик, обозначаемый случайной переменной D. Реализацию этой случайной переменной обозначим символом d. С целью упрощения выкладок, не ограничивая общности, будем рассматривать скалярную модель регрессии. В частности, предполагается, что генерация отклика d определяется следующей вероятностной моделью [525].

1. Из некоторого наперед заданного распределения случайным образом выбирается вектор $\mathbf{x}$.
2. Для заданного вектора $\mathbf{x}$ и некоторого вектора параметров $\mathbf{a}^{(0)}$ выбирается конкретное (например, k-е) правило в соответствии с условной вероятностью $P(k|\mathbf{x},\mathbf{a}^{(0)})$.
3. Для правила $k, k = 1, 2, \ldots, K$, отклик модели является линейным по $\mathbf{x}$ с аддитивной ошибкой ε_k, моделируемой как случайная переменная с гауссовым распределением, имеющим среднее значение нуль и единичную дисперсию, т.е.

$$E[\varepsilon_k] = 0 \text{ для всех } k \tag{7.17}$$

и

$$\text{var}[\varepsilon_k] = 1 \text{ для всех } k. \tag{7.18}$$

Предположение о единичной дисперсии в п. 3 было сделано из соображений дополнительного упрощения изложения. В общем случае каждый эксперт может иметь отличную от 1 дисперсию выходного сигнала, формируемую на основе данных обучения.

Вероятностная генерация переменной D определяется условной вероятностью $P(D = d|\mathbf{x}, \mathbf{w}_k^{(0)})$ для заданного вектора $\mathbf{x}$ и некоторого вектора параметров $\mathbf{w}_k^{(0)}$, где $k = 1, 2, \ldots, K$. Описанная вероятностная порождающая модель не обязательно должна иметь прямую взаимосвязь с некоторым физическим явлением. Требуется лишь, чтобы реализованные в ней вероятностные решения представляли *абстрактную* модель, которая с возрастающей точностью определяет положение *условного среднего значения отклика d в нелинейном многообразии*, которое связывает входной вектор со средним значением выходного сигнала [521].

Согласно представленной модели, отклик D может быть сгенерирован K различными способами в соответствии с K вариантами выбора метки k. Таким образом, условная вероятность генерирования отклика $D = d$ для данного входного вектора $\mathbf{x}$ будет равна

$$P(D = d|\mathbf{x}, \boldsymbol{\theta}^{(0)}) = \sum_{k=1}^{K} P(D = d|\mathbf{x}, \mathbf{w}_k^{(0)}) P(k|\mathbf{x}, \mathbf{a}^{(0)}), \tag{7.19}$$

где $\boldsymbol{\theta}^{(0)}$ — *вектор параметров порождающей модели* (generative model parameter vector), обозначающий комбинацию $\mathbf{a}^{(0)}$ и $\{\mathbf{w}_k^{(0)}\}_{k=1}^{K}$. Верхний индекс 0 в обозначениях $\mathbf{a}^{(0)}$ и $\mathbf{w}_k^{(0)}$ введен для того, чтобы отличать параметры порождающей модели от параметров модели смешения мнений экспертов, которая рассматривается ниже.

Модель смешения мнений экспертов

Рассмотрим конфигурацию сети, показанную на рис. 7.8. Такая сеть носит название *смешения мнений экспертов* (mixture of experts или ME)[4] и состоит из K модулей, обучаемых с учителем и называемых *сетями экспертов* (expert network), или просто *экспертами*. Интегрирующий элемент носит название *сеть шлюза* (gating network). Он выполняет функцию посредника между сетями экспертов. Предполагается, что различные эксперты лучше всего работают в своих областях входного пространства согласно описанной вероятностной порождающей модели. Исходя из этого и возникает потребность в сети шлюза.

В предположении скалярности задачи регрессии каждая из сетей экспертов представляет собой линейный фильтр. На рис. 7.9 показан граф передачи сигнала одного нейрона, соответствующего эксперту k. Таким образом, выходной сигнал, производимый экспертом k, является скалярным произведением входного вектора $\mathbf{x}$ и вектора синаптических весов $\mathbf{w}_k$ данного нейрона, т.е.

$$y_k = \mathbf{w}_k^T \mathbf{x}, \quad k = 1, 2, \ldots, K. \tag{7.20}$$

Сеть шлюза содержит один слой из K нейронов. Каждый из этих нейронов соответствует одному из экспертов. На рис. 7.10, *а* показан архитектурный граф сети шлюза; на рис. 7.10, *б* показан граф передачи сигнала для отдельного нейрона k этой сети. В отличие от экспертов нейроны сети шлюза являются нелинейными. Их функции активации описываются следующим образом:

[4] Идея использования сети экспертов для реализации сложных функций отображений впервые была предложена в [508]. Дальнейшее развитие этой модели было обусловлено предложением, описанным в [789] и рассматривающим конкурентную адаптацию

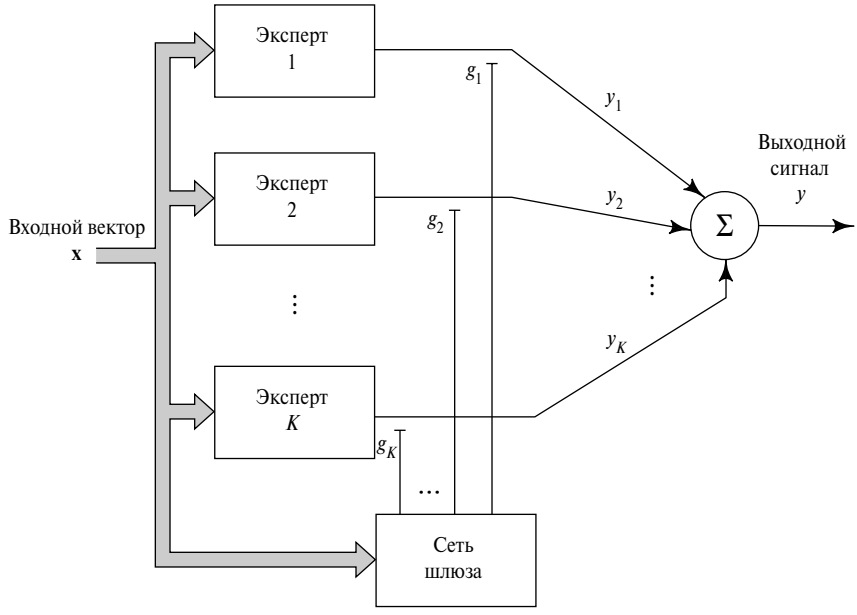

Рис. 7.8. Блочная диаграмма модели ME; скалярные выходы экспертов усреднены сетью шлюза

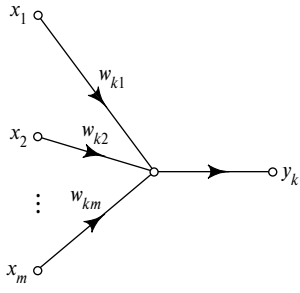

Рис. 7.9. Граф передачи сигнала единичного линейного нейрона, составляющего сеть эксперта k

$$g_k = \frac{\exp(u_k)}{\sum_{j=1}^{K} \exp(u_j)}, \quad k = 1, 2, \ldots, K, \tag{7.21}$$

где u_k — результат скалярного произведения входного вектора $\mathbf{x}$ и вектора синаптических весов $\mathbf{a}_k$, т.е.

$$u_k = \mathbf{a}_k^T \mathbf{x}, \quad k = 1, 2, \ldots, K. \tag{7.22}$$

"Нормализованное" экспоненциальное преобразование (7.21) можно рассматривать как обобщение логистической функции для нескольких входов. В ней сохраня-

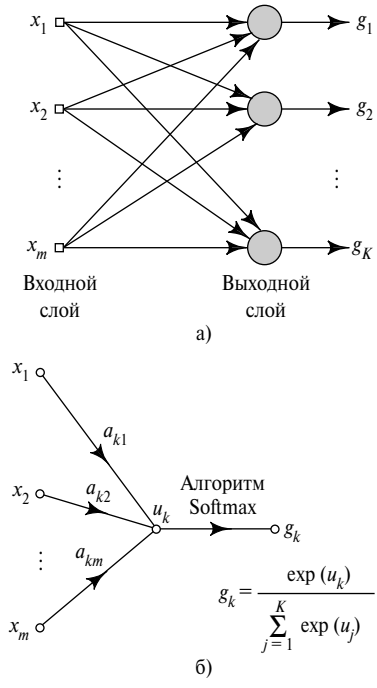

Рис. 7.10. Один слой нейронов сети шлюза (а); граф передачи сигнала для этого нейрона (б)

ется порядок входных значений, но при этом реализовано дифференцируемое обобщение операции “победитель получает все”, извлекающей максимальное значение. По этой причине функция активации (7.21) называется *softmax* [155]. Обратите внимание, что линейная зависимость u_k от входного вектора $\mathbf{x}$ приводит к нелинейной зависимости выходного значения сети шлюза от $\mathbf{x}$.

Для вероятностной интерпретации роли сети шлюза рассмотрим ее как “классификатор”, который отображает входной вектор $\mathbf{x}$ в значение *мультиномиальной вероятности* (multinomial probability) так, чтобы различные эксперты могли соответствовать желаемому отклику [525].

Важно отметить, что использование функции активации softmax в сети шлюза гарантирует, что эти вероятности будут удовлетворять следующим условиям:

$$0 \leq g_k \leq 1 \text{ для всех } k \tag{7.23}$$

и

$$\sum_{k=1}^{K} g_k = 1. \tag{7.24}$$

Пусть y_k — выходной сигнал k-го эксперта, производимый в ответ на входной вектор $\mathbf{x}$. Тогда общий выход модели ME будет следующим:

$$y = \sum_{k=1}^{K} g_k y_k, \tag{7.25}$$

где, как уже отмечалось ранее, g_k — нелинейная функция $\mathbf{x}$. Допуская, что выбрано правило k вероятностной модели и что входным вектором является $\mathbf{x}$, выход эксперта y_k можно трактовать как условное среднее значение случайной переменной D:

$$E[D|\mathbf{x}, k] = y_k = \mathbf{w}_k^T \mathbf{x}, \quad k = 1, 2, \ldots, K. \tag{7.26}$$

Обозначая символом μ_k условное среднее значение переменной D, можно записать:

$$\mu_k = y_k, k = 1, 2, \ldots, K. \tag{7.27}$$

Дисперсия переменной совпадает с дисперсией ошибки ε_k. Таким образом, используя равенство (7.18), приходим к соотношению:

$$\mathrm{var}[D|\mathbf{x}, k] = 1, k = 1, 2, \ldots, K. \tag{7.28}$$

Функция плотности вероятности переменной D для данного входного вектора $\mathbf{x}$ с учетом предположения о том, что выбрано k-е правило вероятностной порождающей модели (т.е. эксперт k), может быть записана в следующем виде:

$$f_D(d|\mathbf{x}, k, \boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(d - y_k)^2\right), \quad k = 1, 2, \ldots, K, \tag{7.29}$$

где $\boldsymbol{\theta}$ — вектор, объединяющий параметры сети шлюза и параметры экспертов модели ME. Функция плотности вероятности переменной D при данном $\mathbf{x}$ является *смесью* функций плотности вероятности $\{f_D(d|\mathbf{x}, k, \boldsymbol{\theta})\}_{k=1}^{K}$, где смешанные параметры являются мультиномиальными вероятностями, определяемыми сетью шлюза. Исходя из этого можно записать следующее:

$$f_D(d|\mathbf{x}, \boldsymbol{\theta}) = \sum_{k=1}^{K} g_k f_D(d|\mathbf{x}, k, \boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi}} \sum_{k=1}^{K} g_k \exp\left(-\frac{1}{2}(d - y_k)^2\right). \tag{7.30}$$

Распределение вероятности (7.30) называется *ассоциативной гауссовой моделью смешения* (associative Gaussian mixture model) [716], [1057], которая вкратце описывалась в главе 5. Ассоциативная модель отличается от неассоциативной тем, что

условное среднее значение μ_k и параметры смешения g_k *не фиксированы*; все они являются функциями входного вектора $\mathbf{x}$. Таким образом, ассоциативная гауссова модель смешения (7.30) может рассматриваться как обобщение обычной гауссовой модели смешения.

Важными чертами модели ME (см. рис. 7.8) при условии ее адекватного обучения являются следующие.

1. Выход y_k k-го эксперта является оценкой условного среднего значения случайной переменной для данного желаемого отклика D, заданного вектора $\mathbf{x}$ и правила k вероятностной порождающей модели.
2. Выход g_k сети шлюза определяет мультиномиальную вероятность того, что выход эксперта k соответствует значению $D = d$ на основе знаний, полученных только от вектора $\mathbf{x}$.

С учетом распределения вероятности (7.30) и заданного множества примеров обучения $\{(\mathbf{x}_i, d_i)\}_{i=1}^N$ задача сводится к *обучению* условного среднего $\mu_k = y_k$ и параметров смешения $g_k, k = 1, 2, \ldots, K$, таким оптимальным образом, чтобы функция $f_D(d|\mathbf{x}, \boldsymbol{\theta})$ представляла собой хорошую оценку функции плотности вероятности среды, отвечающей за генерирование данных обучения.

Пример 7.1

Поверхность регрессии

Рассмотрим модель ME с двумя экспертами и сетью шлюза с двумя выходами, обозначаемыми как g_1 и g_2. Выход g_1 определяется следующей формулой:

$$g_1 = \frac{\exp(u_1)}{\exp(u_1) + \exp(u_2)} = \frac{1}{1 + \exp(-(u_1 - u_2))}. \tag{7.31}$$

Пусть $\mathbf{a}_1$ и $\mathbf{a}_2$ — два вектора весов сети шлюза. Тогда можно записать:

$$u_k = \mathbf{x}^T \mathbf{a}_k, k = 1, 2$$

и, таким образом, переписать выражение (7.31) в следующем виде:

$$g_1 = \frac{1}{1 + \exp(-\mathbf{x}^T(\mathbf{a}_1 - \mathbf{a}_2))}. \tag{7.32}$$

Второй выход сети шлюза можно выразить так:

$$g_2 = 1 - g_1 = \frac{1}{1 + \exp(-\mathbf{x}^T(\mathbf{a}_2 - \mathbf{a}_1))}.$$

Таким образом, оба выхода, g_1 и g_2, имеют форму логистической функции, однако с одним отличием. Ориентация g_1 определяется направлением вектора разности $(\mathbf{a}_1 - \mathbf{a}_2)$, в то время как ориентация g_2 определяется направлением вектора разности $(\mathbf{a}_2 - \mathbf{a}_1)$, т.е. направления векторов g_1 и g_2 противоположны друг другу. Вдоль *хребта* (ridge), определяемого соотношением $\mathbf{a}_1 = \mathbf{a}_2$, имеем $g_1 = g_2 = 1/2$. Таким образом, оба эксперта вносят одинаковый вклад в выход модели ME. В стороне от хребта один из двух экспертов играет доминирующую роль. ■

7.7. Модель иерархического смешения мнений экспертов

Модель ME, изображенная на рис. 7.8, разбивает входное пространство на несколько подпространств. При этом за распределение информации (собранной на основе данных обучения) по отдельным экспертам отвечает одна сеть шлюза. Модель *иерархического смешения мнений экспертов* (hierarchical mixture of experts — HME), представленная на рис. 7.11, представляет собой естественное расширение модели ME. На рисунке показана модель HME с четырьмя экспертами. Архитектура модели HME подобна *дереву*, в котором сети шлюзов являются ветвями, а отдельные эксперты — листьями. Модель HME отличается от модели ME тем, что входное пространство разбивается на множество *вложенных* подпространств, а информация объединяется и перераспределяется между экспертами под управлением нескольких сетей шлюзов, организованных в иерархическую структуру.

Модель HME на рис. 7.11 имеет *два уровня иерархии* (two levels of hierarchy), или *два слоя сетей шлюзов* (two layers of gating networks). Продолжая применять принцип "разделяй и властвуй", способом, аналогичным к проиллюстрированному, можно построить модель HME с несколькими уровнями иерархии. Обратите внимание, что в соответствии с соглашением, принятым на рис. 7.11, нумерация уровней шлюзов начинается от выходного узла дерева.

Модель HME можно формально описать двумя способами [521].

1. *Модель HME является результатом стратегии "разделяй и властвуй"*. Если мы верим, что хорошей стратегией является разбиение входного пространства на области, тогда не менее хорошей стратегией будет деление этих областей на регионы. Эта стратегия может быть продолжена рекурсивно до тех пор, пока мы не достигнем стадии, на которой сложность аппроксимирующих поверхностей не будет соответствовать "локальной" сложности данных обучения. Таким образом, модель HME может работать не хуже, а зачастую и лучше модели ME. И этому факту есть объяснение: более высокий уровень сетей шлюзов модели HME эффективно комбинирует информацию и перераспределяет ее среди экспертов своего поддерева. Следовательно, "сила" каждого из параметров рассматриваемого поддерева используется совместно с другими параметрами, содержащимися в том же поддереве, потенциально улучшая общую производительность модели HME.
2. *Модель HME является деревом мягких решений* (soft-decision). Согласно этой точке зрения, смешение мнений экспертов является всего лишь одноуровневым деревом принятия решений, иногда в шутку называемым *пеньком решений* (decision stump). В более общей постановке модель HME можно рассматривать как вероятностную среду для дерева решений. При этом выходной узел модели HME рассматривается как *корень* дерева. Методология *стандартного* дерева решений

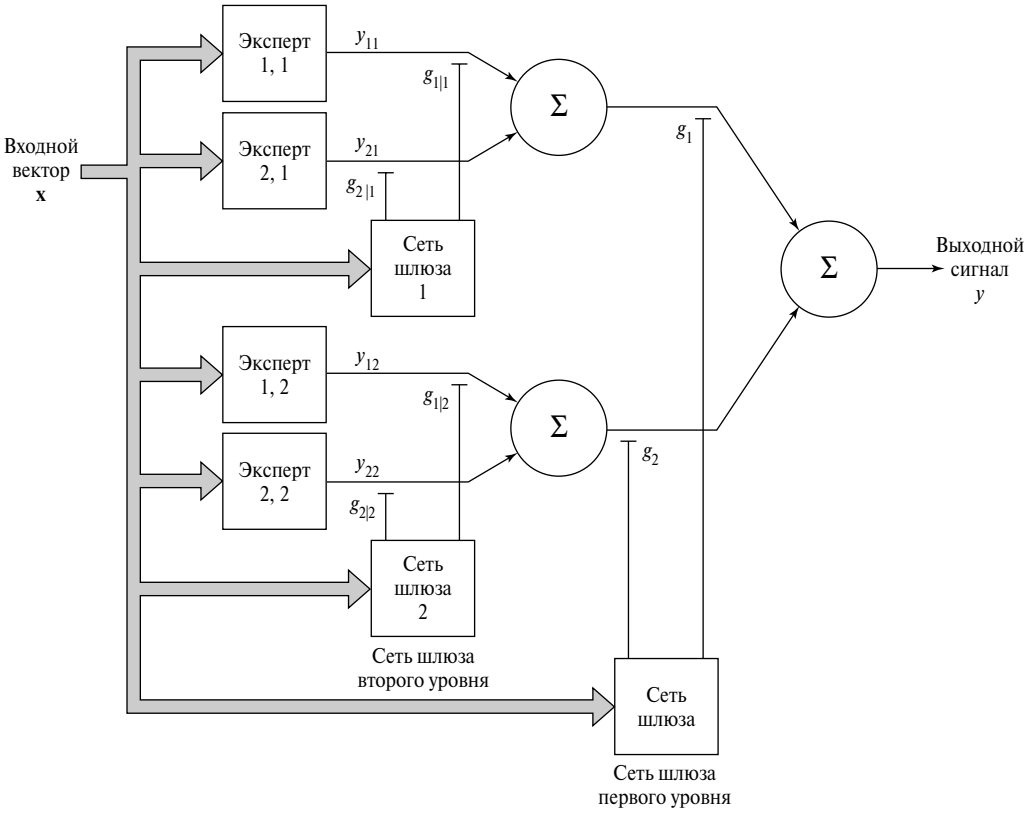

Рис. 7.11. Иерархическое смешение мнений экспертов (HME) для двух уровней иерархии

состоит в построении дерева, ведущего к принятию жестких решений (типа да-нет) в различных областях входного пространства. Противоположностью являются мягкие решения, принимаемые моделью HME. Следовательно, модель HME может превзойти по производительности стандартное дерево решений. Это определяется двумя причинами.

- Жесткие решения, к сожалению, приводят к потере информации, в то время как мягкие решения ее сохраняют. Например, мягкие двоичные решения несут в себе информацию о расстоянии от границы решений (т.е. от точки, в которой решение равновероятно), в то время как жесткие решения — нет. Таким образом, можно утверждать, что, в отличие от стандартного дерева решений, модель обладает встроенным *правилом сохранения информации* (information preservation rule). Это эмпирическое правило утверждает, что информация, содержащаяся во входном сигнале, может быть эффективно сохранена с вычислительной точки зрения до тех пор, пока система не будет готова к окончательному принятию решения или оценке параметра [434].

- Стандартные деревья решений подвержены болезни *жадности* (greediness). Как только в таком дереве принимается решение, оно замораживается и никогда впоследствии не изменяется. Модель HME избавлена от таких проблем, так как решения, принимаемые в дереве, постоянно изменяются. В отличие от стандартного дерева решений в модели HME *можно* отойти от неправильно принятого решения в каком-либо другом месте дерева.

Вторая точка зрения является более предпочтительной. Если рассматривать HME как вероятностный фундамент для построения дерева решений, то можно вычислить подобие для любого предложенного множества данных, а также максимизировать это подобие по параметрам, определяющим разбиение входного пространства на отдельные области. Таким образом, основываясь на информации о стандартных деревьях решений, можно найти практическое решение задачи выбора модели, чем мы и займемся в следующем разделе.

7.8. Выбор модели с использованием стандартного дерева решений

Как и в любой другой нейронной сети, удовлетворительное решение задачи оценки параметров можно получить путем выбора модели, подходящей для конкретной задачи. В случае модели HME под выбором модели понимается выбор количества и композиции узлов решения в дереве. Одно из практических решений задачи выбора модели сводится к запуску на данных обучения алгоритма стандартного дерева решений и принятию полученного дерева в качестве исходного на шаге инициализации в алгоритме обучения, используемом для определения параметров модели HME [521].

Модель HME имеет очевидное сходство со стандартным деревом решений, например с *деревом классификации и регрессии* (classification and regression tree — CART) [154]. На рис. 7.12 показан пример дерева CART, в котором пространство входных данных $\mathbf{X}$ последовательно разбивается серией двоичных делений на *терминальные узлы* (terminal node). При сравнении рис. 7.11 и 7.12 ясно видно, что между моделями CART и HME существует следующее сходство.

- Правило выбора разбиений в промежуточных узлах (ветвях) в модели CART играет роль, аналогичную сетям шлюзов в модели HME.
- Терминальные узлы в модели CART играют роль, аналогичную сетям экспертов в модели HME.

Применяя модель CART для интересующей нас задачи классификации или регрессии, можно использовать преимущества *дискретной* природы этой модели для проведения эффективного поиска среди множества альтернативных деревьев. Используя

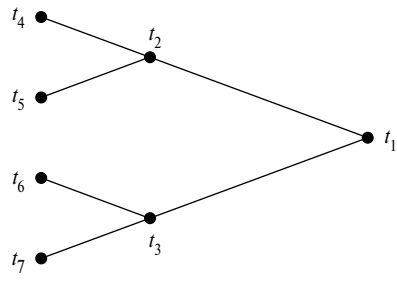

Рис. 7.12. Двоичное дерево решений, описываемое следующим образом: (1) узлы t_2 и t_3 являются потомками узла t_1; (2) узлы t_4 и t_5 являются потомками узла t_2, а узлы t_6 и t_7 являются потомками узла t_3

выбранное таким образом дерево в качестве исходного на шаге инициализации алгоритма обучения, можно взять на вооружение непрерывную вероятностную основу модели HME и получить улучшенную "мягкую" оценку желаемого отклика.

Алгоритм CART

В свете изложенного выше материала приведем краткое описание алгоритма CART. Это описание представлено в контексте задачи регрессии. Имея множество данных обучения $\{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$, алгоритм CART можно использовать для построения двоичного дерева T минимально-квадратичной регрессии. Для этого нужно выполнить следующие действия.

1. *Выбор разбиений* (selection of splits). Пусть узел t — некоторое подмножество дерева T. Пусть $\bar{d}(t)$ — среднее значение d_i для всех пар $(\mathbf{x}_i, d_i)$, попадающих в поддерево t, т.е.

$$\bar{d}(t) = \frac{1}{N(t)} \sum_{\mathbf{x}_i \in t} d_i, \tag{7.33}$$

где суммирование проводится по всем d_i, для которых $\mathbf{x}_i \in t$, а $N(t)$ — общее количество таких случаев. Определим следующие величины:

$$\mathbf{E}(t) = \frac{1}{N} \sum_{\mathbf{x}_i \in t} (d_i - \bar{d}(t))^2 \tag{7.34}$$

и

$$\mathbf{E}(T) = \sum_{t \in T} E(t). \tag{7.35}$$

Для узла t сумма $\sum_{\mathbf{x}_i \in t} (d_i - \bar{d}(t))^2$ представляет собой "внутриузловую сумму квадратов", т.е. общее квадратичное отклонение всех d_i поддерева t от своего среднего

значения $\bar{d}(t)$. Суммирование этих отклонений по всем поддеревьям $t \in T$ дает общую по узлу сумму квадратов, а деление на N — среднее значение.

Для любого заданного множества разбиений S текущего узла t дерева T лучшим разбиением s^* является то, для которого значение $\mathbf{E}(T)$ является наименьшим. Для конкретности предположим, что для любого разбиения s узла t на t_L (новый узел слева от t) и t_R (новый узел справа от t) выполняется соотношение

$$\Delta\mathbf{E}(s,t) = \mathbf{E}(T) - \mathbf{E}(t_L) - \mathbf{E}(t_R). \tag{7.36}$$

Наилучшим разбиением s^* считается то, для которого

$$\Delta\mathbf{E}(s^*,t) = \max_{s\in S} \Delta\mathbf{E}(t,s). \tag{7.37}$$

Таким образом, построенное дерево регрессии будет максимизировать убывание величины $\mathbf{E}(T)$.

2. *Определение терминального узла* (determination of a terminal node). Узел t называется терминальным, если выполняется следующее условие:

$$\max_{s\in S} \Delta\mathbf{E}(t,s) < \beta, \tag{7.38}$$

где β — наперед заданный порог.

3. *Оценка параметров терминального узла по методу наименьших квадратов* (least-squares estimation of a terminal node's parameters). Пусть t является терминальным узлом в двоичном дереве T, а $\mathbf{X}(t)$ — матрица, составленная из $\mathbf{x}_i \in t$. Пусть $\mathbf{d}(t)$ — соответствующий вектор, составленный из желаемых откликов d_i в поддереве t. Определим вектор

$$\mathbf{w}(t) = \mathbf{X}^+(t)\mathbf{d}(t), \tag{7.39}$$

где $\mathbf{X}^+(t)$ — матрица, псевдообратная матрице $\mathbf{X}(t)$. Используя $\mathbf{w}(t)$, на выходе терминального узла t получаем оценку $d(t)$ по методу наименьших квадратов. За счет использования весов, вычисленных согласно (7.39), задача выбора разбиения может быть решена за счет поиска наименьшей суммы квадратов ошибок относительно поверхностей регрессий, а не относительно средних значений.

Использование алгоритма CART для инициализации модели HMA

Предположим, что в результате применения алгоритма CART к множеству данных обучения построено двоичное дерево решений поставленной задачи. Разбиение, построенное алгоритмом CART, можно представить как многомерную поверхность, описываемую формулой

$$\mathbf{a}^T\mathbf{x} + b = 0,$$

где $\mathbf{x}$ — входной вектор; $\mathbf{a}$ — вектор параметров; b — порог.

Рассмотрим в модели HME следующую ситуацию. В примере 7.1 мы обратили внимание, что поверхность регрессии, создаваемая сетью шлюза в двоичном дереве, может быть представлена формулой

$$g = \frac{1}{1 + \exp(-(\mathbf{a}^T\mathbf{x} + b))}, \tag{7.40}$$

описывающей разбиение, в частности при $g = 1/2$. Пусть вектор весов $\mathbf{a}$ для этой конкретной сети шлюза удовлетворяет равенству

$$\mathbf{a} = \|\mathbf{a}\| \cdot \frac{\mathbf{a}}{\|\mathbf{a}\|}, \tag{7.41}$$

где $||\mathbf{a}||$ — длина (Евклидова норма) вектора $\mathbf{a}$; $\mathbf{a}/||\mathbf{a}||$ — нормализованный вектор (имеющий единичную длину). Подставляя (7.41) в (7.40), параметрическое разбиение в сети шлюза можно переписать следующим образом:

$$g = \frac{1}{1 + \exp\left(-\|\mathbf{a}\|\left(\left(\frac{\mathbf{a}}{\|\mathbf{a}\|}\right)^T\mathbf{x} + \frac{b}{\|\mathbf{a}\|}\right)\right)}. \tag{7.42}$$

Отсюда видно, что вектор $\mathbf{a}/||\mathbf{a}||$ задает *направление* разбиения, а $||\mathbf{a}||$ — его *резкость* (sharpness). Важно отметить, что сеть шлюза в выражении (7.42), созданная на основе линейного фильтра, за которым следует нелинейная активационная функция softmax, может имитировать разбиение в стиле CART. Более того, мы получаем дополнительную степень свободы — длину вектора параметров $\mathbf{a}$. В стандартном дереве решений дополнительные параметры не нужны, так как для создания разбиения используется жесткое решение. В отличие от алгоритма ME длина вектора $\mathbf{a}$ оказывает существенное влияние на резкость разбиения, выполняемого сетью шлюза в модели HME. В частности, для вектора синаптических весов $\mathbf{a}$ фиксированного направления можно утверждать следующее.

- Если длина вектора **a** достаточно велика (т.е. при низкой температуре), разбиение оказывается резким (sharp).
- Если вектор **a** имеет малую длину (т.е. при высокой температуре), разбиение является более мягким.

Если в пределе $||\mathbf{a}|| = 0$, то разбиение исчезает и по обе стороны исчезнувшего (мысленного) разбиения$g = 1/2$. Эффект от установки длины вектора **a** в значение нуль равнозначен удалению из дерева нетерминального узла, так как рассматриваемая сеть шлюза больше не будет содержать данного разбиения. В предельном случае, когда вектор **a** крайне мал (т.е. при высокой температуре) во всех нетерминальных узлах, вся модель HMA работает подобно одному узлу, т.е. сводится к обычной модели линейной регрессии (в предположении линейности экспертов). По мере увеличения длины векторов синаптических весов сети шлюза модель HME начинает создавать мягкие разбиения, увеличивая доступное для модели количество степеней свободы.

Таким образом, инициализировать модель HME можно следующим образом.

1. Применить алгоритм CART к данным обучения.
2. Установить векторы синаптических весов экспертов модели HME в значения оценок, полученных методом наименьших квадратов для векторов параметров соответствующих терминальных узлов двоичного дерева, построенного в результате применения алгоритма CART.
3. Для сетей шлюзов:
   - задать векторы синаптических весов в соответствии с направлениями, ортогональными соответствующим разбиениям двоичного дерева, полученным в результате алгоритма CART;
   - установить длины (т.е. Евклидовы нормы) векторов синаптических весов равными значениям длин малых случайных векторов.

7.9. Априорные и апостериорные вероятности

Мультиномиальные (multinomial) вероятности g_k и $g_{j|k}$, относящиеся к сетям шлюзов первого и второго уровней, можно рассматривать как *априорные* вероятности в том смысле, что их значения полностью зависимы от входного вектора **x** (возбудителя). Аналогично, можно определить *апостериорные* вероятности h_k и $h_{j|k}$, значения которых зависят как от входного вектора **x**, так и от выходов экспертов в ответ на сигнал **x**. Это последнее множество вероятностей оказывается полезным при разработке алгоритмов обучения для моделей HME.

Возвращаясь к модели HME на рис. 7.11, можно определить апостериорные вероятности в нетерминальных узлах дерева [526]:

$$h_k = \frac{g_k \sum_{j=1}^{2} g_{j|k} \exp\left(-\frac{1}{2}(d - y_{jk})^2\right)}{\sum_{k=1}^{2} g_k \sum_{j=1}^{2} g_{j|k} \exp\left(-\frac{1}{2}(d - y_{jk})^2\right)} \tag{7.43}$$

и

$$h_{j|k} = \frac{g_{j|k} \exp\left(-\frac{1}{2}(d - y_{jk})^2\right)}{\sum_{j=1}^{2} g_{j|k} \exp\left(-\frac{1}{2}(d - y_{jk})^2\right)}. \tag{7.44}$$

Произведение h_k и $h_{j|k}$ определяет *совместную апостериорную вероятность* (joint a posteriori probability) того, что эксперт (j, k) на выходе даст значение y_{jk}, соответствующее желаемому отклику d, т.е.

$$h_{jk} = h_k h_{j|k} = \frac{g_k g_{j|k} \exp\left(-\frac{1}{2}(d - y_{jk})^2\right)}{\sum_{k=1}^{2} g_k \sum_{j=1}^{2} g_{j|k} \exp\left(-\frac{1}{2}(d - y_{jk})^2\right)}. \tag{7.45}$$

Вероятность h_{jk} удовлетворяет следующим двум условиям:

$$0 \leq h_{jk} \leq 1 \text{ для всех } j \text{ и } k \tag{7.46}$$

и

$$\sum_{j=1}^{2} \sum_{k=1}^{2} h_{jk} = 1. \tag{7.47}$$

Применение (7.47) сводится к тому, чтобы распределять доверие к экспертам на конкурентной основе. Более того, из соотношения (7.45) видно, что чем ближе y_{jk} к d, тем более вероятно, что эксперту (j, k) будет дано доверие на соответствие его выхода желаемому отклику, что интуитивно понятно.

Важным свойством модели HME, заслуживающим особого внимания, является *рекурсивность* вычисления *апостериорной* вероятности. Внимательно изучив выражения (7.43) и (7.44), можно заметить, что знаменатель в (7.44) является частью числителя в (7.43). В модели HME нам требуется вычислить апостериорные вероятности для всех нетерминальных узлов дерева решений. Именно поэтому рекурсивность вычислений приобретает исключительно важное практическое значение. В частности, вычисления апостериорных вероятностей всех нетерминальных узлов дерева можно выполнить за один проход.

- При продвижении по дереву в направлении к корневому узлу, уровень за уровнем, апостериорная вероятность любого нетерминального узла вычисляется как сумма *апостериорных* вероятностей всех его "детей".

7.10. Оценка максимального подобия

Переходя к вопросу оценки параметров модели HME, прежде всего хочется заметить, что ее вероятностная интерпретация слегка отличается от модели ME. Так как модель HME формулируется как двоичное дерево, предполагается, что среда, отвечающая за генерацию данных, включает *вложенную последовательность мягких (двоичных) решений, завершающуюся регрессией входного вектора* $\mathbf{x}$ *к выходному сигналу* d. В частности, предполагается, что в *вероятностной порождающей модели* (probabilistic generative model) HME решения моделируются как мультиномиальные случайные переменные [526]. Это значит, что для каждого входного вектора $\mathbf{x}$ величины $g_i(\mathbf{x}, \boldsymbol{\theta}_i^0)$ интерпретируются как мультиномиальные вероятности, связанные с первым решением, а величины $g_{j|i}(\mathbf{x}, \boldsymbol{\theta}_{ji}^0)$ — как условные мультиномиальные распределения, связанные со вторым решением. Как и ранее, верхний индекс 0 обозначает истинные значения параметров порождающей модели. Эти решения формируют дерево решений. Как и в модели ME, в качестве функции активации всех сетей шлюзов модели HME используется softmax. В частности, функция активации g_k k-го выходного нейрона *сети шлюза верхнего уровня* (top-level gating network) имеет вид

$$g_k = \frac{\exp(u_k)}{\exp(u_1) + \exp(u_2)}, \quad k = 1, 2, \tag{7.48}$$

где u_k — взвешенная сумма входных сигналов, поступающих на данный нейрон. Аналогично, функция активации j-го выходного нейрона в k-й сети шлюза второго уровня иерархии описывается следующим образом:

$$g_{j|k} = \frac{\exp(u_{jk})}{\exp(u_{1k}) + \exp(u_{2k})}, \quad (j, k) = 1, 2, \tag{7.49}$$

где u_{jk} — взвешенная сумма входных сигналов, поступающая на данный нейрон. Из соображений простоты выкладок будем рассматривать модель HME всего с двумя уровнями иерархии (т.е. с двумя слоями сетей шлюзов) (рис. 7.11). Как и для модели ME, предполагается, что каждый из экспертов модели HME состоит из одного слоя линейных нейронов. Пусть y_{jk} — выход эксперта (j, k). Тогда общий выход модели HME можно выразить следующим образом:

$$y = \sum_{k=1}^{2} g_k \sum_{j=1}^{2} g_{j|k} y_{jk}. \tag{7.50}$$

Следуя процедуре, аналогичной описанной в разделе 7.6 для модели ME, можно получить функцию плотности вероятности случайной переменной D, представляющей желаемый отклик модели HME, изображенной на рис. 7.11 для заданного вектора $\mathbf{x}$:

$$f_D(d|\mathbf{x},\boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi}} \sum_{k=1}^{2} g_k \sum_{j=1}^{2} g_{j|k} \exp\left(-\frac{1}{2}(d - y_{jk})^2\right). \tag{7.51}$$

Таким образом, для заданного множества данных обучения выражение (7.51) определяет модель его распределения. Вектор $\boldsymbol{\theta}$ содержит все синаптические веса, характеризующие сети шлюзов и экспертов модели HME.

Роль *функции подобия* (likelihood function), обозначаемой как $l(\boldsymbol{\theta})$, выполняет функция плотности вероятности $f_D(d|\mathbf{x}, \boldsymbol{\theta})$, рассматриваемая как функция вектора параметров $\boldsymbol{\theta}$. Таким образом, можно записать:

$$l(\boldsymbol{\theta}) = f_D(d|\boldsymbol{x},\boldsymbol{\theta}). \tag{7.52}$$

Хотя функция плотности совместной условной вероятности и функция подобия имеют в точности совпадающие формулы, очень важно отметить физическое различие между ними. В случае функции $f_D(d|\mathbf{x}, \boldsymbol{\theta})$ входной вектор $\mathbf{x}$ и вектор параметров $\boldsymbol{\theta}$ фиксированы, а желаемый отклик d является переменной. В случае функции подобия $l(\boldsymbol{\theta})$ фиксированными являются вектор $\mathbf{x}$ и желаемый отклик d; переменным является только вектор параметров $\boldsymbol{\theta}$.

На практике удобнее работать с натуральным логарифмом функции подобия, а не с самой этой функцией. Для обозначения *логарифмической функции подобия* используется обозначение $L(\boldsymbol{\theta})$ вида

$$L(\boldsymbol{\theta}) = \log[l(\boldsymbol{\theta})] = \log f_D[d|\mathbf{x},\boldsymbol{\theta}]. \tag{7.53}$$

Логарифм функции $l(\boldsymbol{\theta})$ является монотонным преобразованием $l(\boldsymbol{\theta})$. Это значит, что при возрастании функции $l(\boldsymbol{\theta})$ ее логарифм $L(\boldsymbol{\theta})$ также возрастает. Так как $l(\boldsymbol{\theta})$ является функцией плотности условной вероятности, она не может принимать отрицательные значения. Отсюда следует, что при вычислении $L(\boldsymbol{\theta})$ не возникнет никаких проблем. Исходя из этого, оценка $\hat{\boldsymbol{\theta}}$ вектора параметров $\boldsymbol{\theta}$ может быть вычислена как решение *уравнения правдоподобия* (likelihood equation):

$$\frac{\partial}{\partial \boldsymbol{\theta}} l(\boldsymbol{\theta}) = \mathbf{0},$$

или, что эквивалентно, *уравнения логарифмического правдоподобия* (log-likelihood equation):

$$\frac{\partial}{\partial \boldsymbol{\theta}} L(\boldsymbol{\theta}) = \mathbf{0}. \tag{7.54}$$

Термин "максимально правдоподобная оценка" (maximum likelihood estimate) с требуемыми асимптотическими свойствами[5] обычно применяется к корню уравнения правдоподобия, который в глобальном смысле максимизирует функцию подобия $l(\boldsymbol{\theta})$. Однако на практике оценка $\hat{\boldsymbol{\theta}}$ может обеспечивать не глобальный, а только локальный максимум. В любом случае оценка максимального правдоподобия, согласно [296], основывается на сравнительно простой идее.

Разные генеральные совокупности (population) генерируют различные данные, при этом происхождение любого заданного примера более правдоподобно для некоторой определенной совокупности, чем для остальных совокупностей.

Более строго, вектор неизвестных параметров $\boldsymbol{\theta}$ для каждого входного вектора $\mathbf{x}$ оценивается своим *наиболее правдоподобным значением* (most plausible value). Другими словами, оценка максимального правдоподобия $\hat{q}$ является тем значением вектора параметров $\boldsymbol{\theta}$, для которой функция плотности условной вероятности $f_D(d|\mathbf{x}, \boldsymbol{\theta})$ является наибольшей.

---

[5] Алгоритмы оценивания максимального правдоподобия имеют ряд привлекательных особенностей. При достаточно общих условиях можно доказать следующие асимптотические свойства [563].

- Алгоритмы оценивания максимального правдоподобия являются согласованными. Пусть $L(\boldsymbol{\theta})$ — функция логарифмического правдоподобия, а θ_i — некоторый элемент вектора параметров $\boldsymbol{\theta}$. Частная производная $\partial L/\partial \theta_i$ называется счетом (score). Утверждается, что алгоритм оценивания максимального правдоподобия является согласованным, в том смысле, что значение θ_i при счете $\partial L/\partial \theta_i$ равном нулю, сходится по вероятности к истинному значению θ_i при стремлении размера используемого при оценке множества примеров к бесконечности.
- Алгоритмы оценивания максимального правдоподобия являются асимптотически эффективными. Это значит, что
  $\lim\limits_{N\to\infty} \left\{ \frac{\mathrm{var}[\theta_i - \hat{\theta}_i]}{I_{ii}} \right\} = 1$ для всех i,
  где N — размер множества обучения; $\hat{\theta}_i$ — максимально правдоподобная оценка θ_i; I_{ii} — i-й диагональный элемент матрицы, обратной к информационной матрице Фишера
  $$\mathbf{J} = -\begin{bmatrix} E\left[\frac{\partial^2 L}{\partial \theta_1^2}\right] & E\left[\frac{\partial^2 L}{\partial \theta_1 \partial \theta_2}\right] & \dots E\left[\frac{\partial^2 L}{\partial \theta_1 \partial \theta_M}\right] \\ E\left[\frac{\partial^2 L}{\partial \theta_2 \partial \theta_1}\right] & E\left[\frac{\partial^2 L}{\partial \theta_2^2}\right] & \dots E\left[\frac{\partial^2 L}{\partial \theta_2 \partial \theta_M}\right] \\ \dots & \dots & \dots \dots \\ E\left[\frac{\partial^2 L}{\partial \theta_M \partial \theta_1}\right] & E\left[\frac{\partial^2 L}{\partial \theta_M \partial \theta_2}\right] & \dots E\left[\frac{\partial^2 L}{\partial \theta_M^2}\right] \end{bmatrix}.$$
  где M — размерность вектора параметров $\boldsymbol{\theta}$.
- Алгоритмы оценивания максимального правдоподобия являются асимптотически гауссовыми. Это значит, что при стремлении размера множества примеров к бесконечности каждый элемент оценки максимального правдоподобия $\hat{\theta}$ имеет гауссово распределение.
  На практике оказывается, что асимптотические (т.е. выполняющиеся при больших размерах множества примеров) свойства алгоритмов оценивания максимального подобия достаточно хорошо выполняются при $N \geq 50$.

7.11. Стратегии обучения для модели HME

Вероятностное описание модели HME в разделе 7.10 привело к построению функции логарифмического подобия $L(\boldsymbol{\theta})$ как объекта максимизации. Остается нерешенным вопрос: как осуществить эту максимизацию. Следует отметить, что не существует единого подхода к решению задач максимизации. Существует несколько различных подходов, два из которых описываются в настоящем разделе [507], [526].

1. Подход на основе стохастического градиента (stochastic gradient approach). Этот подход обеспечивает алгоритм для максимизации $L(\boldsymbol{\theta})$ в реальном времени. Его формулировка для двухуровневой модели (см. рис. 7.11) базируется на следующих формулах.

- Вектор градиента $\partial L/\partial \mathbf{w}_{jk}$ для вектора синаптических весов эксперта (j,k).
- Вектор градиента $\partial L/\partial \mathbf{a}_k$ для вектора синаптических весов выходного нейрона k сети шлюза верхнего уровня.
- Вектор градиента $\partial L/\partial \mathbf{a}_{jk}$ для вектора синаптических весов выходного нейрона сети шлюза второго уровня, связанной с экспертом (j,k).

Достаточно просто показать, что (см. задачу 7.9):

$$\frac{\partial L}{\partial \mathbf{w}_{jk}} = h_{j|k}(n)h_k(n)(d(n) - y_{jk}(n))\mathbf{x}(n), \tag{7.55}$$

$$\frac{\partial L}{\partial \mathbf{a}_k} = (h_k(n) - g_k(n))\mathbf{x}(n), \tag{7.56}$$

$$\frac{\partial L}{\partial \mathbf{a}_{jk}} = h_k(n)(h_{j|k}(n) - g_{j|k}(n))\mathbf{x}(n). \tag{7.57}$$

Равенство (7.55) означает, что в процессе обучения синаптические веса эксперта (j,k) изменяются с целью коррекции ошибки между выходом y_{jk} и желаемым откликом d пропорционально совместной *апостериорной* вероятности h_{jk} того, что выход эксперта (j,k) соответствует желаемому отклику d. Равенство (7.56) утверждает, что синаптические веса выходного нейрона k сети шлюза верхнего уровня настраиваются так, чтобы *априорные* вероятности $g_k(n)$ смещались в сторону соответствующих *апостериорных* вероятностей $h_k(n)$. Равенство (7.57) означает, что синаптические веса выходного нейрона сети шлюза второго уровня, связанной с экспертом (j,k), настраиваются с целью коррекции ошибки между *априорной* вероятностью $g_{j|k}$ и соответствующей *апостериорной* вероятностью $h_{j|k}$ пропорционально *апостериорной* вероятности $h_k(n)$.

В соответствии с равенствами (7.55)–(7.57) синаптические веса модели HME изменяются после представления ей каждого примера (возбуждения). Суммируя векторы градиентов по n, можно сформулировать пакетную версию метода градиентного спуска для максимизации функции логарифмического подобия $L(\boldsymbol{\theta})$.

2. Подход на основе максимизации ожидания (expectation-maximization approach — EM). Алгоритм максимизации ожидания (EM), согласно [252], реализует итеративную процедуру вычисления оценки максимального правдоподобия в ситуациях, когда эта проблема решается очень легко. Алгоритм EM получил свое название из-за того, что каждая итерация алгоритма состоит из двух шагов.

- *Шаг ожидания* (expectation step), на котором множество наблюдаемых данных *неполной задачи* (incomplete data problem) и текущее значение вектора параметров используются для получения расширенного *полного набора данных* (complete data set).
- *Шаг максимизации* (maximizing step) заключается в вычислении новой оценки вектора параметров путем максимизации функции логарифмического подобия полного множества данных, созданного на первом шаге.

Таким образом, начиная с подходящего значения вектора параметров, эти действия повторяются поочередно до полной сходимости.

Алгоритм EM оказывается полезным не только в тех случаях, когда набор данных явно не полон, но также и во многих других ситуациях, в которых неполнота данных является не столь очевидной или естественной. В самом деле, вычисление оценки максимального правдоподобия часто облегчается ее формулировкой как задачи с неполными данными. Это происходит потому, что алгоритм EM позволяет использовать пониженную сложность оценки максимального правдоподобия при неполных данных [717]. Модель HME является одним из таких примеров применения. В данном случае отсутствующие данные в виде некоторых переменных-индикаторов искусственно вводятся в модель HME для построения оценки максимального правдоподобия неизвестного вектора параметров (см. раздел 7.12).

Важными особенностями модели HME (при любом подходе — максимизации ожидания или стохастического градиента) являются следующие.

- Все сети шлюзов в модели последовательно вычисляют апостериорные вероятности для всех точек данных множества примеров.
- Корректировки синаптических весов экспертов и сетей шлюзов модели при переходе к следующей итерации являются функциями апостериорной вероятности и соответствующей ей априорной вероятности.

Следовательно, если сеть эксперта, расположенная ниже по дереву, не справляется с задачей обобщения данных в своей локальной окрестности, то происходит смещение регрессионной (дискриминантной) поверхности сети шлюза, расположенной выше в этом дереве. В свою очередь, это смещение помогает экспертам на следующей итерации алгоритма обучения лучше аппроксимировать (fit) данные путем смещения соответствующих подпространств данных. За счет этого модель HME улучшает эффективность “жадного” алгоритма решения, свойственного стандартному дереву решений, например CART.

7.12. Алгоритм EM

Алгоритм EM занимает особое место благодаря своей простоте и общности лежащей в его основе теории, а также широкой области применения[6]. В этом разделе представлено описание алгоритма EM в общем виде. В последующих разделах мы продолжим рассмотрение его применения в задаче оценки параметров модели HME.

Пусть вектор $\mathbf{z}$ состоит из отсутствующих или ненаблюдаемых данных, а $\mathbf{r}$ — вектор с полными данными, составленный из наблюдений d и вектора отсутствующих данных $\mathbf{z}$. Таким образом, существуют два пространства — $\mathbf{D}$ и $\mathbf{R}$, а также отображение пространства $\mathbf{R}$ в $\mathbf{D}$ типа "много к одному". Однако вместо наблюдения полного вектора данных $\mathbf{r}$, фактически можно наблюдать только неполные данные $d = d(\mathbf{r})$ в пространстве $\mathbf{D}$.

Пусть $f_c(\mathbf{r}|\boldsymbol{\theta})$ — функция плотности условной вероятности $\mathbf{r}$ для данного вектора параметров $\boldsymbol{\theta}$. Отсюда следует, что функция плотности условной вероятности случайной переменной D для данного вектора $\boldsymbol{\theta}$ определяется следующим образом:

$$f_D(d|\boldsymbol{\theta}) = \int_{\mathbf{R}(d)} f_c(\mathbf{r}|\,\boldsymbol{\theta}) d\mathbf{r}, \tag{7.58}$$

где $\mathbf{R}(d)$ — подпространство $\mathbf{R}$, определяемое равенством $d = d(\mathbf{r})$. Алгоритм EM направлен на поиск значения $\boldsymbol{\theta}$, максимизирующего *функцию логарифмического правдоподобия на неполных данных* (incomplete-data log-likelihood function):

$$L(\boldsymbol{\theta}) = \log f_D(d|\boldsymbol{\theta}).$$

Эта задача косвенно решается итеративным применением *функции логарифмического правдоподобия на полных данных* (complete-data log-likelihood function):

$$L_c(\boldsymbol{\theta}) = \log f_c(\mathbf{r}|\boldsymbol{\theta}), \tag{7.59}$$

которая является случайной переменной ввиду того, что отсутствующий вектор данных $\mathbf{z}$ не известен.

---

[6] Работу, рассматривающую оценку параметров смеси двух инвариантных гауссовых распределений [779], можно считать самой ранней литературной ссылкой по теме процессов типа EM.

Название "EM-алгоритм" было введено в [252]. В этой фундаментальной работе впервые была представлена формулировка алгоритма EM для вычисления оценок максимального правдоподобия на неполных множествах данных с различными уровнями общности.

Первый объединенный доклад по теории, методологии и применениям алгоритма EM, по его истории и расширениям был представлен в сборнике, выпущенном в 1997 году [717].

Более строго, пусть $\hat{\boldsymbol{\theta}}(n)$ — значение вектора параметров $\boldsymbol{\theta}$ на итерации n алгоритма EM. На E-шаге этой итерации вычисляем математическое ожидание

$$Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}(n)) = E[L_c(\boldsymbol{\theta})] \tag{7.60}$$

по $\hat{\boldsymbol{\theta}}(n)$. На M-шаге этой же итерации вычисляем максимум функции $Q(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}(n))$ по $\boldsymbol{\theta}$ в пространстве параметров (весов) $\mathbf{W}$ и, таким образом, находим измененную оценку вектора параметров $\hat{\boldsymbol{\theta}}(n+1)$:

$$\hat{\theta}(n+1) = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}(n)). \tag{7.61}$$

Этот алгоритм начинается с некоторого начального значения $\hat{\boldsymbol{\theta}}(0)$ вектора параметров $\boldsymbol{\theta}$. Затем оба шага последовательно повторяются в соответствии с выражениями (7.60) и (7.61) соответственно, пока разность между $L(\hat{\boldsymbol{\theta}}(n+1))$ и $L(\hat{\boldsymbol{\theta}}(n))$ не окажется меньше некоторого малого наперед заданного значения. В этой точке работа алгоритма завершается.

Заметим, что после каждой итерации алгоритма EM функция логарифмического правдоподобия на неполных данных *не* убывает, т.е. (см. задачу 7.10):

$$L(\hat{\boldsymbol{\theta}}(n+1)) \quad \geq \quad L(\hat{\boldsymbol{\theta}}(n)) \text{ для } n = 0, 1, \ldots. \tag{7.62}$$

Выполнение равенства в этой формуле означает, что мы находимся в стационарной точке функции логарифмического правдоподобия[7].

7.13. Применение алгоритма EM к модели HME

Ознакомившись с алгоритмом EM, можно приступать к решению задачи оценивания параметров модели HME[8].

---

[7] При достаточно общих условиях подобные значения, вычисленные по алгоритму EM, сходятся к стационарным значениям. В [1167] проводится детальное исследование свойств сходимости алгоритма EM. Однако этот алгоритм *не всегда* приводит к локальному или глобальному максимуму функции правдоподобия. В главе 3 [717] представлены два примера, подтверждающие этот факт. В первом из них алгоритм сходился к седловой точке, а во втором — к локальному *минимуму* функции правдоподобия.

[8] С помощью алгоритма EM можно также вычислить байесовскую апостериорную оценкумаксимума (Bayesian maximum a posterior estimation) за счет использования априорной информации для вектора параметров (см. задачу 7.11). Используя правило Байеса, функцию плотности условной вероятности вектора параметров $\boldsymbol{\theta}$ для заданного множества наблюдений $\mathbf{x}$ можно выразить так:

$f_{\boldsymbol{\theta}}(\boldsymbol{\theta}|\mathbf{x}) = \frac{f_{\mathbf{x}}(\mathbf{x}|\boldsymbol{\theta}) f_{\boldsymbol{\theta}}(\boldsymbol{\theta})}{f_{\mathbf{x}}(\mathbf{x})}$.

Из этого соотношения ясно видно, что максимизация апостериорной плотности $f_\theta(\boldsymbol{\theta},\mathbf{x})$ эквивалентна максимизации произведения $f_{\mathbf{x}}(\mathbf{x}|\boldsymbol{\theta}) f_{\boldsymbol{\theta}}(\boldsymbol{\theta})$, так как $f_{\mathbf{x}}(\mathbf{x})$ не зависит от $\boldsymbol{\theta}$. Функция плотности вероятности $f_\theta(\boldsymbol{\theta})$ представляет собой доступную априорную информацию о $\boldsymbol{\theta}$. Максимизация $f_\theta(\boldsymbol{\theta},\mathbf{x})$ дает наиболее вероятную оценку вектора параметров $\boldsymbol{\theta}$ для данного вектора $\mathbf{x}$. В контексте этой оценки важно подчеркнуть следующие моменты.

Пусть $g_k^{(i)}$ и $g_{j|k}^{(i)}$ — (условные) мультиномиальные вероятности, соответствующие решениям, принимаемым сетями шлюзов первого уровня k и второго уровня (j,k) соответственно (см. рис. 7.11), при обработке примера i из обучающего множества. Тогда из выражения (7.51) видно, что соответствующие значения функций плотности условной вероятности случайной переменной D для данного примера $\mathbf{x}_i$ и вектора параметров $\boldsymbol{\theta}$ задаются следующей формулой:

$$f_D(d_i|\mathbf{x}_i,\boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi}} \sum_{k=1}^{2} g_k^{(i)} \sum_{j=1}^{2} g_{j|k}^{(i)} \exp\left(-\frac{1}{2}(d_i - y_{jk}^{(i)})^2\right), \tag{7.63}$$

где $y_{jk}^{(i)}$ — выходной сигнал эксперта (j,k) в ответ на подачу i-го примера из множества обучения. Предположим, что все N примеров, содержащихся в обучающем множестве, являются статистически независимыми. Тогда можно определить функцию логарифмического подобия для задачи с неполными данными следующего вида:

$$L(\boldsymbol{\theta}) = \log\left[\prod_{i=1}^{N} f_D(d_i|\mathbf{x}_i,\boldsymbol{\theta})\right]. \tag{7.64}$$

Подставляя выражение (7.63) в (7.64) и игнорируя константу $-(1/2)\log(2\pi)$, получим:

$$L(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log\left[\sum_{k=1}^{2} g_k^{(i)} \sum_{j=1}^{2} g_{j|k}^{(i)} \exp\left(-\frac{1}{2}(d_i - y_{jk}^{(i)})^2\right)\right]. \tag{7.65}$$

Чтобы вычислить оценку максимального правдоподобия для вектора $\boldsymbol{\theta}$, необходимо найти стационарную точку (т.е. локальный или глобальный максимум) функции $L(\boldsymbol{\theta})$. К сожалению, функция логарифмического подобия $L(\boldsymbol{\theta})$, согласно формуле (7.65), еще не подходит для такого рода вычислений.

Чтобы обойти эту сложность, *искусственно* дополним множество данных наблюдений $\{d_i\}_{i=1}^N$ множеством отсутствующих данных, полученных в результате выполнения алгоритма EM. Это можно сделать путем введения *переменных-индикаторов* (indicator variable), относящихся к вероятностной модели архитектуры HME, следующим образом [526].

Оценка максимального правдоподобия, представленная максимизацией $f_{\boldsymbol{\theta}}(\boldsymbol{\theta},\mathbf{x})$ по $\boldsymbol{\theta}$, является сокращенной формой максимизации апостериорной вероятности (под словом “сокращенная” понимается отсутствие априорной информации).

Использование априорной информации является синонимом регуляризации, которая (см. главу 5) соответствует гладкому отображению входа на выход.

В [1114] представлен байесовский подход к оценке параметров модели смеси экспертов, преодолевающий так называемое явление избыточного обучения, которое приводит к оценке с большой дисперсией при использовании принципа максимума подобия.

- $z_k^{(i)}$ и $z_{j|k}^{(i)}$ будем интерпретировать как метки соответствующие решениям, принимаемым вероятностной моделью в ответ на подачу i-го примера множества обучения. Эти переменные определяются таким образом, чтобы при любом значении i только одно из значений $z_k^{(i)}$ и только одно из значений $z_{j|k}^{(i)}$ равнялось единице. Обе переменные — $z_k^{(i)}$ и $z_{j|k}^{(i)}$ — можно рассматривать как статистически независимые дискретные случайные переменные, математическое ожидание которых равно соответственно

$$E[z_k^{(i)}] = P\left[z_k^{(i)} = 1|\mathbf{x}_i, d_i, \hat{\boldsymbol{\theta}}(n)\right] = h_k^{(i)} \tag{7.66}$$

и

$$E[z_{j|k}^{(i)}] = P\left[z_{j|k}^{(i)} = 1|\mathbf{x}_i, d_i, \hat{\boldsymbol{\theta}}(n)\right] = h_{j|k}^{(i)}, \tag{7.67}$$

где $\hat{\boldsymbol{\theta}}(n)$ — оценка параметра $\boldsymbol{\theta}$ на итерации n алгоритма EM.

- $z_{jk}^{(i)} = z_{j|k}^{(i)} z_k^{(i)}$ интерпретируется как метка, определяющая эксперта (j,k) в вероятностной модели для i-го примера множества обучения. Ее также можно трактовать как дискретную случайную величину со следующим математическим ожиданием:

$$E[z_{jk}^{(i)}] = E[z_{j|k}^{(i)} z_k^{(i)}] = E[z_{j|k}^{(i)}]E[z_k^{(i)}] = h_{j|k}^{(i)} h_k^{(i)} = h_{jk}^{(i)}. \tag{7.68}$$

Значения $h_k^{(i)}, h_{j|k}^{(i)}$ и $h_{jk}^{(i)}$ в равенствах (7.66) и (7.68) являются апостериорными вероятностями, введенными в разделе 7.9. Верхний индекс i введен для идентификации рассматриваемого примера в обучающем множестве. В задаче 7.13 читателю будет предложено обосновать все эти три равенства.

С добавлением определенных таким образом отсутствующих данных к множеству наблюдений задача вычисления оценки максимального правдоподобия существенно упрощается. Пусть $f_c(d_i, z_{jk}^{(i)}|\mathbf{x}_i, \boldsymbol{\theta})$ — функция плотности условной вероятности на полном множестве данных, составленная для d_i и $z_{jk}^{(i)}$ при заданных $\mathbf{x}_i$ и векторе параметров $\boldsymbol{\theta}$. Тогда можно записать:

$$f_c(d_i, z_{jk}^{(i)}|\mathbf{x}_i, \boldsymbol{\theta}) = \prod_{j=1}^{2}\prod_{k=1}^{2}(g_k^{(i)} g_{j|k}^{(i)}(i) f_{jk}(d_i)), \tag{7.69}$$

где $f_{jk}(d_i)$ — функция плотности условной вероятности d_i для выбранного эксперта (j,k) в модели HME, т.е. $f_{jk}(i)$ выбирается из гауссова распределения

$$f_{jk}(d_i) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(d_i - y_{jk}^{(i)})^2\right). \tag{7.70}$$

Обратите внимание, что формула (7.69) соответствует гипотетическому эксперименту, содержащему переменные-индикаторы, представленные переменными $z_{jk}^{(i)}$, не существующими в физическом смысле. В любом случае функция логарифмического правдоподобия для задачи с полными данными, включающими в себя все множество примеров обучения, выглядит следующим образом:

$$\begin{aligned} L_c(\boldsymbol{\theta}) &= \log\left[\prod_{i=1}^{N} f_c(d_i, z_{jk}^{(i)}|\mathbf{x}_i, \boldsymbol{\theta})\right] = \\ &= \log\left[\prod_{i=1}^{N}\prod_{j=1}^{2}\prod_{k=1}^{2}\left(g_k^{(i)} g_{j|k}^{(i)}(i) f_{jk}(d_i)\right)^{z_{jk}^{(i)}}\right] = \\ &= \sum_{i=1}^{N}\sum_{j=1}^{2}\sum_{k=1}^{2} z_{jk}^{(i)}\left[\log g_k^{(i)} + \log g_{j|k}^{(i)} + \log f_{jk}(d_i)\right]. \end{aligned} \tag{7.71}$$

Подставляя выражение (7.70) в (7.71) и игнорируя константу –(1/2)log(2π), можно записать:

$$L_c(\boldsymbol{\theta}) = \sum_{i=1}^{N}\sum_{j=1}^{2}\sum_{k=1}^{2} z_{jk}^{(i)}\left[\log g_k^{(i)} + \log g_{j|k}^{(i)} - \frac{1}{2}(d_i - y_{jk}^{(i)})^2\right]. \tag{7.72}$$

Сравнивая выражения (7.72) и (7.65), можно заметить, что введение в множество наблюдений переменных-индикаторов вместо отсутствующих данных дало следующий вычислительный эффект: задача вычисления оценки максимального правдоподобия была разделена на множество задач регрессии для отдельных экспертов и отдельное множество задач классификации для сетей шлюзов.

Для продолжения работы с алгоритмом EM нужно в первую очередь использовать E-шаг этого алгоритма, вычисляя математическое ожидание функции логарифмического подобия на полных данных $L_c(\boldsymbol{\theta})$:

$$\begin{aligned} Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}(n)) &= E[L_c(\boldsymbol{\theta})] = \\ &= \sum_{i=1}^{N}\sum_{j=1}^{2}\sum_{k=1}^{2} E\left[z_{jk}^{(i)}\right] \cdot \left[\log g_k^{(i)} + \log g_{j|k}^{(i)} - \frac{1}{2}(d_i - y_{jk}^{(i)})^2\right], \end{aligned} \tag{7.73}$$

где оператор ожидания действует на переменную индикатора $z_{jk}^{(i)}$ как на единственную ненаблюдаемую переменную. Тогда, подставляя (7.68) в (7.73), получим:

$$Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}(n)) = \sum_{i=1}^{N} \sum_{j=1}^{2} \sum_{k=1}^{2} h_{jk}^{(i)} \cdot \left[\log g_k^{(i)} + \log g_{j|k}^{(i)} - \frac{1}{2}(d_i - y_{jk}^{(i)})^2 \right]. \tag{7.74}$$

М-шаг этого алгоритма требует максимизации функции $Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}(n))$ по $\boldsymbol{\theta}$. Вектор параметров $\boldsymbol{\theta}$ состоит из двух множеств синаптических весов: одно из них относится к сетям шлюзов, второе — к экспертам. Из предыдущего обсуждения можно сделать следующие выводы.

- Синаптические веса экспертов определяют $y_{jk}^{(i)}$, которые также входят в определение $h_{jk}^{(i)}$. Таким образом, на выражение для $Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}(n))$ эксперты влияют только в части слагаемого $h_{jk}^{(i)}(d_i - y_{jk}^{(i)})^2$.
- Синаптические веса сетей шлюзов определяют вероятности $g_{jk}^{(i)}, g_{j|k}^{(i)}$ и $h_{jk}^{(i)}$. Таким образом, на выражение для $Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}(n))$ эксперты влияют только в части слагаемого $h_{jk}^{(i)}(\log g_k^{(i)} + \log g_{j|k}^{(i)})$.

Следовательно, М-шаг алгоритма EM сводится к следующей тройственной задаче оптимизации модели HME с двумя уровнями иерархии:

$$\mathbf{w}_{jk}(n+1) = \arg\min_{\mathbf{w}_{jk}} \sum_{i=1}^{N} h_{jk}^{(i)}(d_i - y_{jk}^{(i)})^2, \tag{7.75}$$

$$\mathbf{a}_j(n+1) = \arg\min_{\mathbf{a}_j} \sum_{i=1}^{N} \sum_{k=1}^{2} h_{jk}^{(i)} \log g_k^{(i)} \tag{7.76}$$

и

$$\mathbf{a}_{jk}(n+1) = \arg\min_{\mathbf{a}_{jk}} \sum_{i=1}^{N} \sum_{l=1}^{2} h_l^{(i)} \sum_{m=1}^{2} h_{m|l}^{(i)} \log g_{m|l}^{(i)}. \tag{7.77}$$

Задачи оптимизации (7.75)–(7.77) решаются при фиксированных h; h является функцией параметров, но производные по h не вычисляются. Обратите внимание, что величины в правых частях выражений относятся к измерениям, выполняемым на шаге n.

Оптимизация выражения (7.75), относящегося к экспертам, представляет собой задачу нахождения взвешенной оценки по методу наименьших квадратов. Остальные две задачи оптимизации — (7.76) и (7.77) — относятся к сетям шлюзов и представляют собой задачи поиска оценки максимального правдоподобия[9].

Несмотря на то что уравнения сформулированы только для двух уровней иерархии, они могут быть расширены для любого их количества.

[9] Для решения задачи вычисления оценки максимального правдоподобия, описываемой уравнениями (7.76) и (7.77), часто используют алгоритм, известный под названием итеративно взвешиваемый алгоритм наименьших квадратов

7.14. Резюме и обсуждение

При изучении задач моделирования, распознавания образов и регрессии необходимо рассматривать два экстремальных случая.

1. *Простые модели*, которые дают представление о данной задаче, но не отличаются особой точностью.
2. *Сложные модели*, которые дают точные результаты, не слишком углубляясь в саму задачу.

Похоже, в рамках одной модели невозможно объединить простоту и точность. В контексте дискуссии, представленной во второй части настоящей главы, алгоритм CART является примером простой модели, которая использует жесткие решения при построении разбиения входного пространства на множество подпространств, каждое из которых имеет собственного эксперта. К сожалению, использование результатов жестких решений приводит к потере части информации и, таким образом, потерям производительности. С другой стороны, многослойный персептрон (MLP) является сложной моделью с вложенной формой нелинейности, создаваемой для сохранения информации из множества обучающих данных. Однако в нем используется подход "черного ящика" для глобального построения единой функции аппроксимации обучающих данных. При этом теряется глубина взгляда на саму задачу. Модель HME, представляющая собой динамический тип ассоциативной машины, обладая преимуществами как CART, так и MLP, является компромиссом между этими двумя экстремальными случаями.

- Архитектура HME аналогична архитектуре CART, но отличается от нее мягкостью разбиения входного пространства.
- HME использует вложенную форму нелинейности, аналогичную MLP, но не для построения отображения входного сигнала в выходной, а с целью разбиения входного пространства.

В этой главе мы заострили внимание на использовании двух методов построения модели HME.

- Алгоритм CART составляет архитектурную основу для задачи выбора модели.
- Алгоритм EM используется для решения задачи оценки параметров. Он итеративно вычисляет оценки максимального правдоподобия для параметров модели.

Алгоритм EM обычно обеспечивает движение вверх по склону функции правдоподобия. Таким образом, инициализируя алгоритм EM с помощью CART так, как описано в разделе 7.8, можно ожидать лучшего возможного качества обобщения первого из них, применяя результаты второго в качестве исходного состояния.

Алгоритм EM является важным и основополагающим в тех областях, где на первое место выходит построение оценки максимального правдоподобия, например в *моделировании*. Интересное приложение моделирования было описано в [509], где модель ME обучалась решению задачи "что–где". В этой задаче модель должна определить, что представляет собой объект и где он находится в визуальном поле. При обучении использовались два эксперта, каждый из которых специализировался на одном из аспектов задачи. Для заданного входного сигнала оба эксперта генерировали выходные. При этом сеть шлюза принимала решение относительно смешения выходов. Успешные результаты, полученные в этой работе, показали, что задачи можно естественным образом распределять, но не на основе принципа "одна задача на всех", а на основе соответствия требований задачи вычислительным свойствам модели [282].

В завершение обсуждения вернемся к еще одному классу ассоциативных машин, о котором говорилось в первой части настоящей главы. В то время как модели HME и ME основаны на использовании сетей шлюзов, активизируемых входными сигналами для объединения знаний, накопленных разными экспертами модели, ассоциативные машины основаны на усреднении по ансамблю или, как альтернатива, на усилении, основанном на интеграции самим алгоритмом обучения.

1. Усреднение по ансамблю улучшает качество обучения в смысле снижения ошибки (error performance), мудро комбинируя два эффекта.
   - Уменьшение уровня ошибки, вводимой порогом, при помощи целенаправленного избыточного обучения отдельных экспертов ассоциативной машины.
   - Уменьшение уровня ошибки, создаваемой дисперсией, за счет использования различных начальных состояний при обучении различных экспертов, и последующего усреднения по ансамблю их выходных сигналов.
2. Усиление улучшает эффективность алгоритма собственным оригинальным способом. В данном случае от отдельных экспертов требуется качество, лишь слегка отличающееся в лучшую сторону от случайного выбора. Слабая обучаемость экспертов преобразуется в сильную, при этом ошибку ассоциативной машины можно сделать сколь угодно малой. Эта выдающаяся метаморфоза достигается за счет *фильтрации* распределения входных данных, приводящей к тому, что слабо обучаемые модели (т.е. эксперты) в конечном итоге обучаются на всем распределении, либо за счет создания *подвыборки* множества обучения в соответствии с некоторым распределением, как в алгоритме AdaBoost. Преимущество последнего по сравнению с фильтрацией состоит в том, что он работает с обучающим множеством фиксированного размера.

Задачи

Усреднение по ансамблю

7.1. Рассмотрим ассоциативную машину, состоящую из K экспертов. Функция отображения входа на выход k-го эксперта обозначается через $F_k(\mathbf{x})$, где $\mathbf{x}$ — входной вектор, а $k = 1, 2, \ldots, K$. Выходы отдельных экспертов линейно суммируются, формируя общий выход системы y, следующим образом:

$$y = \sum_{k=1}^{K} w_k F_k(\mathbf{x}),$$

где w_k — линейный вес, назначенный $F_k(\mathbf{x})$. Требуется оценить w_k так, чтобы выход y обеспечивал оценку желаемого отклика d, соответствующего $\mathbf{x}$ по методу наименьших квадратов. Имея множество данных обучения $\{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$, определите искомые значения w_k для решения этой задачи оценки параметров.

Усиление

7.2. Сравните вычислительные преимущества и недостатки методов усиления за счет фильтрации и AdaBoost.

7.3. Обычно усиление лучше всего выполнять на слабых моделях обучения, т.е. на моделях с относительно низким уровнем ошибок обобщения. Однако предположим, что имеется некоторая сильная модель обучения, т.е. модель с относительно высоким уровнем ошибок обобщения. Предположим также, что имеется множество примеров фиксированного размера. Как в такой ситуации выполнить усиление за счет фильтрации или алгоритм AdaBoost?

Смешение мнений экспертов

7.4. Рассмотрим следующую кусочно-линейную задачу:

$$F(x_1, x_2, \ldots, x_{10}) = \begin{cases} 3x_2 + 2x_3 + x_4 + 3 + \varepsilon, & \text{если } x_1 = 1, \\ 3x_5 + 2x_6 + x_7 - 3 + \varepsilon, & \text{если } x_1 = -1. \end{cases}$$

Для сравнения используются следующие конфигурации сети.

1. Многослойный персептрон: сеть $10 \to 10 \to 1$.
2. Смешение мнений экспертов: сеть шлюза: $10 \to 2$. Сеть эксперта: $10 \to 1$.

Сравните вычислительную сложность этих двух сетей.

7.5. Модель ME, описываемая функцией плотности условной вероятности (7.30), основана на модели скалярной регрессии, в которой ошибка имеет гауссово распределение со средним значением нуль и единичной дисперсией.

а) Переформулируйте это уравнение для более общего случая модели ME, соответствующей модели множественной регрессии, в которой желаемый отклик является вектором размерности q, а ошибка — многомерным гауссовым распределением с нулевым средним значением и матрицей ковариации $\boldsymbol{\sigma}$.

б) Чем эта переформулированная модель отличается от модели ME, показанной на рис. 7.8?

7.6. Выведите алгоритм стохастического градиента для обучения модели смеси экспертов.

Иерархическое смешение мнений экспертов

7.7. а) Постройте блочную диаграмму модели HME с тремя уровнями иерархии. Предполагается, что для данной модели используется двоичное дерево решений.

б) Запишите апостериорные вероятности для нетерминальных узлов модели, описанной в пункте а). Продемонстрируйте рекурсивность вычислений, проводимых при оценке этих вероятностей.

в) Определите функцию плотности условной вероятности для модели HME, описанной в пункте а).

7.8. Обсудите сходства и отличия между моделью HME и сетями на основе радиально-базисных функций.

7.9. Выведите уравнения, описывающие алгоритм стохастического градиента для обучения модели HME с двумя уровнями иерархии. Предполагается, что для этой модели используется двоичное дерево решений.

Алгоритм EM и его применение в модели HME

7.10. Докажите свойство монотонного возрастания алгоритма EM, описываемого уравнением (7.62). Для этого выполните следующее.

а) Пусть

$$k(\mathbf{r}|d,\boldsymbol{\theta}) = \frac{f_c(\mathbf{r}|\boldsymbol{\theta})}{f_D(d|\boldsymbol{\theta})}$$

обозначает функцию плотности условной вероятности расширенного вектора параметров **r**, для данного наблюдения d и вектора параметров $\boldsymbol{\theta}$. Исходя из этого, функция логарифмического правдоподобия на неполных данных будет иметь следующий вид:

$$L(\boldsymbol{\theta}) = L_c(\boldsymbol{\theta}) - \log k(\boldsymbol{r}|d, \theta),$$

где $L_c(\theta) = \log f_c(\mathbf{r}|\theta)$ — функция логарифмического правдоподобия на полных данных. Зная математическое ожидание функции $L(\theta)$ относительно условного распределения **r** и значение d, покажите, что

$$L(\boldsymbol{\theta}) = Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}(n)) - K(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}(n)),$$

где

$$K(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}(n)) = E[\log k(\mathbf{r}|d, \hat{\boldsymbol{\theta}})].$$

При этих условиях покажите, что

$$L(\hat{\boldsymbol{\theta}}(n+1)) - L(\hat{\boldsymbol{\theta}}(n)) = \left[Q(\hat{\boldsymbol{\theta}}(n+1), \hat{\boldsymbol{\theta}}(n)) - Q(\hat{\boldsymbol{\theta}}(n), \hat{\boldsymbol{\theta}}(n))\right] - \\ - \left[K(\hat{\boldsymbol{\theta}}(n+1), \hat{\boldsymbol{\theta}}(n)) - K(\hat{\boldsymbol{\theta}}(n), \hat{\boldsymbol{\theta}}(n))\right].$$

б) *Неравенство Йенсена* (Jensen's inequality) утверждает, что если функция $g(\cdot)$ является выпуклой, а u — некоторая случайная переменная, то

$$E[g(u)] \leq g(E[u]),$$

где E — оператор математического ожидания. Более того, если $g(\cdot)$ — строго выпуклая функция, то из равенства в этом соотношении с вероятностью 1 следует, что $u = E(u)$ [221].

Используя неравенство Йенсена, покажите, что

$$K(\hat{\boldsymbol{\theta}}(n+1), \hat{\boldsymbol{\theta}}(n)) - K(\hat{\boldsymbol{\theta}}(n), \hat{\boldsymbol{\theta}}(n)) \leq 0. \tag{7.78}$$

Исходя из этого, покажите, что уравнение (7.62) выполняется для $n = 0, 1, 2, \ldots$.

7.11. Алгоритм EM довольно легко модифицируется для включения максимума апостериорной оценки вектора параметров $\boldsymbol{\theta}$. Используя правило Байеса, модифицируйте шаги E и M этого алгоритма, чтобы обеспечить эту оценку.

7.12. Если модель HME, обучаемая с помощью алгоритма EM, и многослойный персептрон, обучаемый по алгоритму обратного распространения, имеют одинаковые показатели производительности для данной задачи, интуитивно ожидается, что вычислительная сложность первого будет превосходить сложность второго. Приведите аргументы за или против этого утверждения.

7.13. Обоснуйте связь между переменными-индикаторами и соответствующими апостериорными вероятностями, описанными в формулах (7.66) и (7.68).

7.14. Уравнение (7.75) описывает взвешенный метод наименьших квадратов для оптимизации сетей экспертов модели HME, представленной на рис. 7.11, в предположении, что желаемый отклик d — скаляр. Как модифицировать это соотношение для случая многомерного желаемого отклика?