Содержание

7 Разреженные ядерные методы

В предыдущей главе мы рассмотрели алгоритмы машинного обучения, основанные на нелинейных ядрах. Многие из этих алгоритмов имеют существенный недостаток — ядро $k(\mathbf{x}_n, \mathbf{x}_m)$ должно быть вычислено для всех возможных пар $\mathbf{x}_n$ и $\mathbf{x}_m$ обучающих точек, что может оказаться невозможным во время обучения и приводить к чрезмерно долгим вычислениям при прогнозировании для новых точек. В этой главе будут рассмотрены *разреженные ядерные алгоритмы*, у которых прогнозы для новых входов зависят только от ядра, вычисленного на подмножестве обучающих точек.

Начнем изложение с подробного описания *метода опорных векторов* (Support Vector Machines — SVM), который несколько лет назад стал популярным методом решения задач классификации, регрессии и обнаружения новизны. Важным свойством метода опорных векторов является то, что определение параметров модели сводится к задаче выпуклой оптимизации, поэтому любое локальное решение также является глобальным оптимумом. Поскольку метод

опорных векторов широко использует метод множителей Лагранжа, читателю предлагается ознакомиться с ключевыми понятиями, приведенными в ***приложении Д***. Дополнительная информация о методе опорных векторов приведена в Vapnik (1995), Burges (1998), Cristianini и Shawe-Taylor (2000), Müller *et al.* (2001), Schölkopf и Smola (2002) и Herbrich (2002).

SVM — это метод принятия решений, и поэтому в нем не предусмотрено вычисление апостериорных вероятностей. Мы уже обсуждали некоторые из преимуществ вычисления вероятностей в ***разделе 1.5.4***. Альтернативный метод разреженных ядер, известный как *метод релевантных векторов* (relevance vector machine — RVM), основан на байесовском выводе и предусматривает вычисление апостериорных вероятностей для выходов, а также, как правило, порождает гораздо более разреженные решения, чем SVM.

7.1. Методы классификации с максимальным зазором

Начнем обсуждение метода опорных векторов, вернувшись к задаче бинарной классификации с помощью линейных моделей вида

$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}) + b, \tag{7.1}$$

где $\boldsymbol{\phi}(\mathbf{x})$ обозначает фиксированное преобразование пространства признаков, и мы сделали явным параметр смещения b. Заметим, что вскоре мы сформулируем двойственное представление, выраженное в терминах ядра, что позволяет не работать явным образом в пространстве признаков. Обучающее множество содержит N входных векторов $\mathbf{x}_1, \ldots, \mathbf{x}_N$ с соответствующими целевыми значениями $t_1, \ldots, t_N$, где $t_n \in \{-1, 1\}$, а новые точки $\mathbf{x}$ классифицируются по знаку числа $y(\mathbf{x})$.

Предположим пока, что обучающее множество является линейно разделимым в пространстве признаков, так что по определению существует по крайней мере один набор параметров $\mathbf{w}$ и b, такой, что функция вида (7.1) удовлетворяет условию $y(\mathbf{x}_n) > 0$ для точек с $t_n = +1$ и $y(\mathbf{x}_n) < 0$ для точек, удовлетворяющих условию $t_n = -1$, так что $t_n y(\mathbf{x}_n) > 0$ для всех обучающих точек.

Конечно, существует множество таких решений, которые точно разделяют классы. В ***разделе 4.1.7*** описан алгоритм персептрона, который гарантированно найдет решение за конечное количество шагов. Однако решение, которое он находит, будет зависеть от (произвольных) начальных значений, выбранных для $\mathbf{w}$ и b, а также от порядка, в котором представлены обучающие точки. Если существует несколько решений, которые точно классифицируют обучающее множество, то мы должны попытаться найти то из них, которое дает наименьшую

ошибку обобщения. Метод опорных векторов подходит к этой задаче с помощью концепции *зазора* (margin), который определяется как наименьшее расстояние между границей решения и любой из выборок (рис. 7.1).

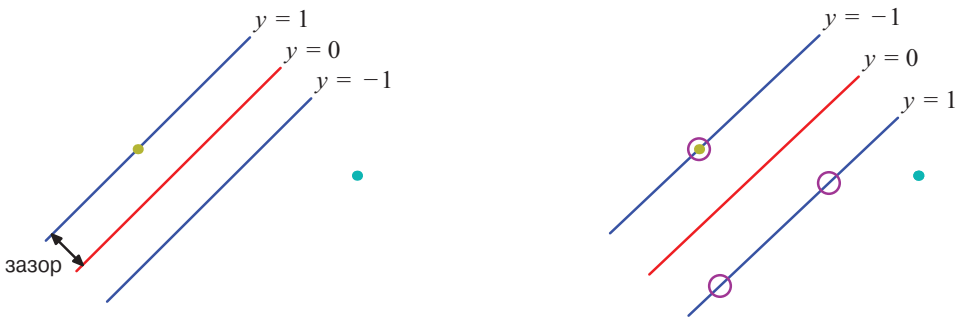


Рис. 7.1. Зазор — это перпендикулярное расстояние между границей решения и ближайшей точкой исходных данных, как показано на рисунке слева. Максимизация зазора приводит к определенному выбору границы решения, как показано справа. Местоположение этой границы определяется подмножеством точек исходных данных, называемых опорными векторами, которые обозначены кружками

В методе опорных векторов граница решения выбирается так, чтобы зазор был максимальным. Решение о максимальном зазоре может быть обосновано с помощью *теории вычислительного обучения*, также известной как *теория статистического обучения* (***см. раздел 7.1.5***). Тем не менее простое понимание истоков концепции максимального зазора было дано Tong and Koller (2000), которые рассмотрели теорию классификации, основанную на гибриде порождающих и дискриминационных подходов. Сначала они моделируют распределение по входным векторам $\mathbf{x}$ для каждого класса с использованием оценки плотности Парзена с гауссовыми ядрами, имеющими общий параметр σ^2. Вместе с априорным распределением класса это позволяет определить оптимальную границу принятия решения с минимальным уровнем ошибок. Однако, вместо того чтобы использовать эту оптимальную границу, они определяют лучшую гиперплоскость, минимизируя вероятность ошибки относительно построенной модели плотности. В пределе при $\sigma^2 \to 0$ оптимальная гиперплоскость обеспечивает максимальный зазор. Интуитивные представления, лежащие в основе этого результата, заключаются в том, что при уменьшении σ^2 точки, расположенные близко к гиперплоскости, доминируют над более отдаленными. В пределе гиперплоскость становится независимой от точек, которые не являются опорными векторами.

На рис. 10.13 мы увидим, что в байесовском подходе маргинализация по априорному распределению параметров для простого линейно разделимого множества данных приводит к границе решения, которая лежит в середине области, разделяющей точки. Аналогичным свойством обладает решение с большим зазором.

Напомним (см. рис. 4.1), что кратчайшее расстояние от точки $\mathbf{x}$ до гиперплоскости, определяемой уравнением $y(\mathbf{x}) = 0$, где функция $y(\mathbf{x})$ имеет вид (7.1), задается формулой $|y(\mathbf{x})|/\|\mathbf{w}\|$. Кроме того, нас интересуют только решения, для которых все точки исходных данных классифицированы правильно, так что $t_n y(\mathbf{x}_n) > 0$ для всех n. Таким образом, расстояние от точки $\mathbf{x}_n$ до поверхности решения определяется формулой

$$\frac{t_n y(\mathbf{x}_n)}{\|\mathbf{w}\|} = \frac{t_n\left(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)+b\right)}{\|\mathbf{w}\|}. \tag{7.2}$$

Зазор задается кратчайшим расстоянием до ближайшей точки $\mathbf{x}_n$ от множества данных, и мы хотим найти такие параметры $\mathbf{w}$ и $\mathbf{b}$, чтобы максимизировать это расстояние. Таким образом, решение с максимальным зазором является решением задачи

$$\arg\max_{\mathbf{w},b}\left\{\frac{1}{\|\mathbf{w}\|}\min_n\left[t_n\left(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)+b\right)\right]\right\}, \tag{7.3}$$

где множитель $1/\|\mathbf{w}\|$ не оптимизируется по n, так как $\mathbf{w}$ не зависит от n. Прямое решение этой задачи оптимизации было бы очень сложным, поэтому мы преобразуем ее в эквивалентную задачу, которую намного легче решить. Для этого заметим, что если мы выполним масштабирование $\mathbf{w} \to \kappa\mathbf{w}$ и $b \to \kappa b$, то расстояние от любой точки $\mathbf{x}_n$ до поверхности решения, определенного величиной $t_n y(\mathbf{x}_n)/\|\mathbf{w}\|$, не изменится. Мы можем использовать эту возможность и установить условие

$$t_n(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) + b) = 1 \tag{7.4}$$

для точки, ближе всего расположенной к поверхности. В этом случае все точки исходных данных будут удовлетворять ограничениям

$$t_n(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) + b) \geq 1,\ n = 1,\ldots, N. \tag{7.5}$$

Это выражение называется каноническим представлением гиперплоскости решения. Для точек, на которых выполняется равенство, ограничения считаются *активными*, в для остальных они — *неактивными*. По определению всегда будет существовать хотя бы одно активное ограничение, потому что всегда найдется

точка, самая близкая к гиперплоскости, и после максимизации зазора будет существовать как минимум два активных ограничения. Тогда задача оптимизации просто сводится к максимизации $\|\mathbf{w}\|^{-1}$, что эквивалентно минимизации $\|\mathbf{w}\|^2$, поэтому мы должны решить задачу оптимизации

$$\arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \tag{7.6}$$

при ограничениях (7.5). Множитель 1/2 в (7.6) включен для удобства, которое проявится в дальнейшем. Это пример задачи *квадратичного программирования*, в которой мы пытаемся минимизировать квадратичную функцию, подчиненную набору ограничений в виде линейных неравенств. Может показаться, что параметр смещения b исчез из оптимизации. Тем не менее он определяется неявно через ограничения, поскольку они требуют, чтобы изменения в $\|\mathbf{w}\|$ были компенсированы изменениями в b. Вскоре мы увидим, как это работает.

Для решения этой задачи оптимизации с ограничениями введем множители Лагранжа $a_n \geq 0$ с одним множителем a_n для каждого ограничения в (7.5), построив функцию Лагранжа (***см. приложение Д***)

$$L(\mathbf{w},b,\mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n \left\{ t_n \left(\mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n) + b \right) - 1 \right\}, \tag{7.7}$$

где $\mathbf{a} = (a_1, \ldots, a_N)^{\mathrm{T}}$. Обратите внимание на знак “минус” перед множителем Лагранжа, поскольку мы минимизируем по $\mathbf{w}$ и b и максимизируем по $\mathbf{a}$. Приравнивая производные от $L(\mathbf{w}, b, \mathbf{a})$ по $\mathbf{w}$ и b к нулю, получим следующие два условия:

$$\mathbf{w} = \sum_{n=1}^{N} a_n t_n \boldsymbol{\phi}(\mathbf{x}_n), \tag{7.8}$$

$$0 = \sum_{n=1}^{N} a_n t_n. \tag{7.9}$$

Исключение $\mathbf{w}$ и b из $L(\mathbf{w}, b, \mathbf{a})$ с использованием этих условий дает двойственную формулировку задачи о максимальном зазоре, в которой мы максимизируем функцию

$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \tag{7.10}$$

при ограничениях

$$a_n \geq 0, \quad n = 1, \ldots, N, \tag{7.11}$$

$$\sum_{n=1}^{N} a_n t_n = 0. \tag{7.12}$$

Здесь ядро определяется как $k(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}')$. Как и в предыдущем случае, задача принимает форму задачи квадратичного программирования, в которой мы оптимизируем квадратичную функцию при ограничениях в виде неравенств. Мы обсудим методы решения таких задач квадратичного программирования в ***разделе 7.1.1***.

Решение задачи квадратичного программирования при M переменных в общем случае имеет вычислительную сложность порядка $O(M^3)$. При переходе к двойственной формулировке мы преобразовали исходную задачу оптимизации, которая предусматривала минимизацию (7.6) по M переменным, в двойственную задачу (7.10), которая имеет N переменных. Для фиксированного набора базисных функций, количество которых M меньше количества точек исходных данных N, переход к двойственной задаче оказывается невыгодным. Однако он позволяет переформулировать модель с использованием ядер, и поэтому метод классификации с максимальным зазором может эффективно применяться к пространствам признаков, размерность которых превышает количество точек исходных данных, включая бесконечные пространства признаков. В формулировке с использованием ядра также разъясняется роль ограничения, которое состоит в том, что ядро $k(\mathbf{x}, \mathbf{x}')$ положительно определено и, следовательно, функция Лагранжа $\tilde{L}(\mathbf{a})$ ограничена сверху, т.е. задача оптимизации определена корректно.

Чтобы классифицировать новые точки с помощью обученной модели, мы оцениваем знак $y(\mathbf{x})$, определенный формулой (7.1). Это можно выразить в терминах параметров $\{a_n\}$ и ядра, подставив вместо $\mathbf{w}$ выражение (7.8):

$$y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b. \tag{7.13}$$

В ***приложении Д*** показано, что задача оптимизации с ограничениями такого вида удовлетворяет *условиям Каруша–Куна–Таккера* (Karush–Kuhn–Tucker — KKT), которые в этом случае требуют, чтобы выполнялись следующие три свойства:

$$a_n \geq 0, \tag{7.14}$$

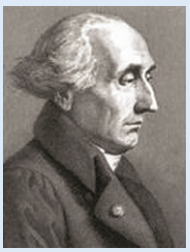
$$t_n y(\mathbf{x}_n) - 1 \geq 0, \tag{7.15}$$

$$a_n \{t_n y(\mathbf{x}_n) - 1\} = 0. \tag{7.16}$$

Таким образом, для каждой точки данных либо $a_n = 0$, либо $t_n y(\mathbf{x}_n) = 1$. Любая точка данных, для которой $a_n = 0$, не будет учитываться в сумме (7.13) и, следовательно, не играет никакой роли в создании прогнозов для новых точек.

Остальные точки исходных данных называются *опорными векторами*, и поскольку они удовлетворяют условиям $t_n y(\mathbf{x}_n) = 1$, они соответствуют точкам, которые лежат на гиперплоскостях с максимальным зазором в пространстве признаков, как показано на рис. 7.1. Это свойство является основным для практического применения метода опорных векторов. После обучения модели значительная часть точек исходных данных может быть отброшена и сохранены только опорные векторы.

Джозеф-Луи Лагранж
1736–1813



Хотя Лагранж и считается французским математиком, он родился в Турине (Италия). В возрасте девятнадцати лет он уже внес важный вклад в математику и был назначен профессором Королевской артиллерийской школы в Турине. В течение многих лет Эйлер упорно убеждал Лагранжа переехать в Берлин, что он в конце концов и сделал в 1766 году, когда он стал преемником Эйлера на посту директора математического департамента Берлинской академии. Позже он переехал в Париж, чудом оставшись в живых во время французской революции благодаря личному вмешательству Лавуазье (французского химика, открывшего кислород), который сам впоследствии был казнен на гильотине. Лагранж внес важный вклад в вариационное исчисление и основы динамики.

Решив задачу квадратичного программирования и найдя значение для $\mathbf{a}$, мы можем затем определить значение порогового параметра b, заметив, что любой опорный вектор $\mathbf{x}_n$ удовлетворяет условию $t_n y(\mathbf{x}_n) = 1$. Используя (7.13), получим:

$$t_n\left(\sum_{m\in\mathcal{S}} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) + b\right) = 1, \tag{7.17}$$

где $\mathcal{S}$ — множество индексов опорных векторов. Хотя мы можем решить это уравнение относительно b, используя произвольно выбранный опорный вектор $\mathbf{x}_n$, с вычислительной точки зрения более устойчивое решение получается, если мы сначала умножим уравнение на t_n, с учетом, что $t_n^2 = 1$, а затем усредним эти уравнения по всем опорным векторам и решим относительно b:

$$b = \frac{1}{N_{\mathcal{S}}} \sum_{n \in \mathcal{S}} \left(t_n - \sum_{m \in \mathcal{S}} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) \right), \tag{7.18}$$

где $N_{\mathcal{S}}$ — общее количество опорных векторов.

Для последующего сравнения с альтернативными моделями мы можем выразить классификацию с максимальным зазором в терминах минимизации функции ошибок с помощью простого квадратичного регуляризатора в виде

$$\sum_{n=1}^{N} E_{\infty}\left(y(\mathbf{x}_n) t_n - 1\right) + \lambda \|\mathbf{w}\|^2, \tag{7.19}$$

где $E_{\infty}(z)$ — функция, равная нулю, если $z \geq 0$, и ∞ в противном случае. Эта функция гарантирует, что ограничения (7.5) выполнены. Заметим, что до тех пор, пока параметр регуляризации удовлетворяет условию $\lambda > 0$, его точное значение не играет никакой роли.

На рис. 7.2 приведен пример классификации, полученный в результате обучения метода опорных векторов на простом искусственном множестве данных, использующем гауссово ядро в виде (6.23). Хотя множество данных не является линейно разделимым в двумерном пространстве исходных данных $\mathbf{x}$, оно линейно разделяется в нелинейном пространстве признаков, неявно определяемом нелинейным ядром. Таким образом, обучающие точки отлично разделяются в пространстве исходных данных.

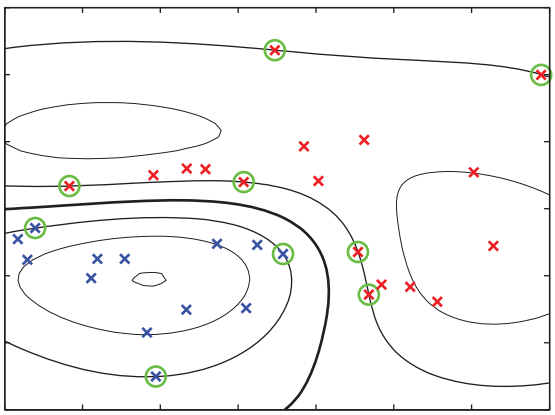


Рис. 7.2. Пример искусственных данных из двух классов в двух измерениях, показывающих контуры постоянных значений $y(\mathbf{x})$, полученных по методу опорных векторов с гауссовым ядром. Также показаны границы решения, границы зазоров и опорные векторы

Этот пример также дает геометрическое представление о причинах разреженности в методе SVM. Гиперплоскость максимального зазора определяется расположением опорных векторов. Другие точки исходных данных могут свободно перемещаться (пока они остаются за пределами зазора), никак не влияя на границы решения, и поэтому решение не будет зависеть от таких точек.

7.1.1. Перекрытие распределений классов

До сих пор мы предполагали, что точки обучающего множества линейно разделимы в пространстве признаков $\boldsymbol{\phi}(\mathbf{x})$. Полученный метод опорных векторов даст точное разделение обучающих данных в пространстве исходных данных $\mathbf{x}$, хотя соответствующая граница решения будет нелинейной. Однако на практике условные по классу распределения могут перекрываться, и в этом случае точное разделение обучающих данных может привести к плохому обобщению.

Следовательно, нам нужен способ изменения метода опорных векторов, допускающий неправильную классификацию некоторых обучающих точек. Из (7.19) видно, что в случае разделимых классов мы неявно использовали функцию ошибок, которая давала бесконечную ошибку, если точка данных была классифицирована ошибочно, и нулевую ошибку, если она была классифицирована правильно, а затем оптимизировали параметры модели для максимизации зазора. Теперь мы изменим этот подход, чтобы точкам данных разрешалось находиться на неправильной стороне от границы зазора, но со штрафом, который увеличивается с расстоянием от этой границы. Для последующей оптимизации удобно сделать это штраф линейной функцией этого расстояния. Для этого введем *фиктивные переменные* $\xi_n \geq 0$, где $n = 1, ..., N$, с одной фиктивной переменной для каждой точки обучающих данных (Bennett, 1992; Cortes and Vapnik, 1995). Они определены как $\xi_n = 0$ для точек исходных данных, которые находятся на правильной стороне (на границе или внутри области), и $\xi_n = |t_n - y(\mathbf{x})|$ для других точек. Таким образом, точка данных, находящаяся на границе решения $y(\mathbf{x}_n) = 0$, будет иметь $\xi_n = 1$, а точки с $\xi_n > 1$ будут классифицированы ошибочно. Затем точные ограничения классификации (7.5) заменяются на неравенства

$$t_n y(\mathbf{x}_n) \geq 1 - \xi_n, \quad n = 1, ..., N, \tag{7.20}$$

в которых фиктивные переменные должны удовлетворять условию $\xi_n \geq 0$. Точки данных, для которых $\xi_n = 0$, классифицированы правильно и находятся либо на границе, либо на правильной стороне зазора. Точки, для которых $0 < \xi_n \leq 1$ лежат внутри зазора, но на правильной стороне границы решения, а те точки данных, для которых $\xi_n > 1$, лежат на неправильной стороне границы решения и

классифицируются ошибочно, как показано на рис. 7.3. Иногда это явление описывается как ослабление жестких ограничений, чтобы создать *мягкий зазор* и позволяет некорректно классифицировать некоторые точки обучающих данных. Обратите внимание, что хотя фиктивные переменные допускают перекрывающиеся распределения классов, эта структура по-прежнему чувствительна к выбросам, потому что штраф за ошибочную классификацию линейно возрастает в зависимости от ξ_n.

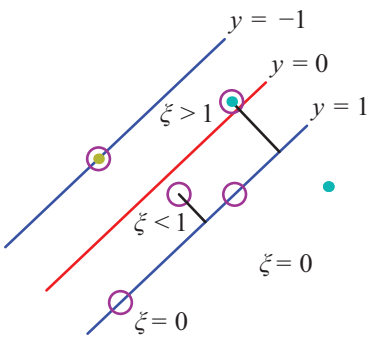


Рис. 7.3. Фиктивные переменные $\xi_n \geq 0$. Точки данных с кружками вокруг них являются опорными векторами

Наша цель состоит в том, чтобы максимизировать зазор, мягко штрафуя точки, которые лежат на неправильной стороне от границы зазора. Следовательно, мы минимизируем функцию

$$C\sum_{n=1}^{N}\xi_n + \frac{1}{2}\|\mathbf{w}\|^2, \tag{7.21}$$

где параметр $C > 0$ управляет компромиссом между штрафом фиктивной переменной и зазором. Поскольку любая точка, которая ошибочно классифицирована, удовлетворяет условию $\xi_n > 1$, число $\sum_n \xi_n$ является верхней границей количества ошибочных точек. Таким образом, параметр C аналогичен (обратному) коэффициенту регуляризации, поскольку он контролирует компромисс между минимизацией ошибок обучения и контролем сложности модели. В пределе при $C \to \infty$ мы получаем предыдущий вариант метода опорных векторов для разделяемых данных.

Теперь мы хотим минимизировать (7.21) с учетом ограничений (7.20) вместе с $\xi_n \geq 0$. Соответствующий лагранжиан задается формулой

$$L(\mathbf{w},b,\boldsymbol{\xi},\mathbf{a},\boldsymbol{\mu}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n - \sum_{n=1}^{N}a_n\{t_n y(\mathbf{x}_n) - 1 + \xi_n\} - \sum_{n=1}^{N}\mu_n\xi_n, \tag{7.22}$$

где $\{a_n \geq 0\}$ и $\{\mu \geq 0\}$ — множители Лагранжа. Соответствующее множество условий KKT задается формулами (***см. приложение Д***)

$$a_n \geq 0, \tag{7.23}$$
$$t_n y(\mathbf{x}_n) - 1 + \xi_n \geq 0, \tag{7.24}$$
$$a_n(t_n y(\mathbf{x}_n) - 1 + \xi_n) \geq 0, \tag{7.25}$$
$$\mu_n \geq 0, \tag{7.26}$$
$$\xi_n \geq 0, \tag{7.27}$$
$$\mu_n \xi_n = 0, \tag{7.28}$$

где $n = 1, ..., N$.

Теперь оптимизируем $\mathbf{w}$, b и $\{\xi_n\}$, используя определение (7.1) функции $y(\mathbf{x})$:

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{n=1}^{N} a_n t_n \boldsymbol{\phi}(\mathbf{x}_n), \tag{7.29}$$

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \quad \Rightarrow \quad \sum_{n=1}^{N} a_n t_n = 0, \tag{7.30}$$

$$\frac{\partial L}{\partial \xi_n} = 0 \quad \Rightarrow \quad a_n = C - \mu_n. \tag{7.31}$$

Используя эти результаты для исключения $\mathbf{w}$, b и $\{\xi_n\}$ из лагранжиана, получим двойственный лагранжиан в виде

$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m), \tag{7.32}$$

который идентичен случаю линейно разделимых множеств, за исключением того, что ограничения несколько отличаются. Чтобы увидеть, каковы эти ограничения, отметим, что условие $a_n \geq 0$ требуется, потому что a_n — это множители Лагранжа. Кроме того, из (7.31) и условия $\mu_n \geq 0$ следует $a_n \leq C$. Следовательно, нужно максимизировать (7.32) по двойственным переменным $\{a_n\}$:

$$0 \leq a_n \leq C, \tag{7.33}$$

$$\sum_{n=1}^{N} a_n t_n = 0 \tag{7.34}$$

для $n = 1, ..., N$, где (7.33) известны как *квадратные ограничения* (box constraints). Эта задача снова представляет собой задачу квадратичного программирования. Если подставить (7.29) в (7.1), мы увидим, что предсказания для новых точек снова вычисляются по формуле (7.13).

Теперь мы можем интерпретировать полученное решение. Как и ранее, некоторые точки исходных данных могут удовлетворять условию $a_n = 0$, и в этом случае они не вносят вклад в прогностическую модель (7.13). Остальные точки являются опорными векторами. Они удовлетворяют условию $a_n > 0$ и, как следует из (7.25), должны удовлетворять условию

$$t_n y(\mathbf{x}_n) = 1 - \xi_n. \tag{7.35}$$

Если $a_n < C$, то из (7.31) следует, что $\mu_n > 0$, а с учетом (7.28) это значит, что $\xi_n = 0$ и, следовательно, такие точки лежат на границе. Точки, удовлетворяющие условию $a_n = C$, могут лежать внутри зазора и быть правильно классифицированными, если $\xi_n \leq 1$, или ошибочно классифицированными, если $\xi_n > 1$.

Для определения параметра b в (7.1) отметим, что те опорные векторы, для которых $0 < a_n < C$, удовлетворяют условию $\xi_n = 0$, так что $t_n y(\mathbf{x}_n) = 1$, и, следовательно, будут удовлетворять условию

$$t_n\left(\sum_{m\in\mathcal{S}} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) + b\right) = 1. \tag{7.36}$$

Как и в предыдущем случае, вычислительно устойчивое решение получается путем усреднения:

$$b = \frac{1}{N_{\mathcal{M}}}\sum_{n\in\mathcal{M}}\left(t_n - \sum_{m\in\mathcal{S}} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m)\right), \tag{7.37}$$

где $\mathcal{M}$ обозначает набор индексов точек исходных данных, удовлетворяющих условию $0 < a_n < C$.

Альтернативная, но эквивалентная формулировка метода опорных векторов, известная как ν-SVM, была предложена Schölkopf *et al.* (2000). Он предполагает максимизацию

$$\tilde{L}(\mathbf{a}) = -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \tag{7.38}$$

с учетом ограничений

$$0 \leq a_n \leq 1/N, \tag{7.39}$$

$$\sum_{n=1}^{N} a_n t_n = 0, \tag{7.40}$$

$$\sum_{n=1}^{N} a_n \geq \nu. \tag{7.41}$$

Преимущество этого подхода состоит в том, что параметр ν, который заменяет C, можно интерпретировать и как верхнюю границу доли *ошибок зазора* (точек, удовлетворяющих условию $\xi_n > 0$ и, следовательно, лежащих на неправильной стороне границы зазора и допускающих как правильную, так и неправильную классификацию), и как нижнюю границу доли опорных векторов. Пример ν-SVM, примененный к искусственному набору данных, приведен на рис. 7.4. Здесь использовались гауссовы ядра вида $\exp(-\gamma\|\mathbf{x}-\mathbf{x}'\|^2)$ с $\gamma = 0{,}45$.

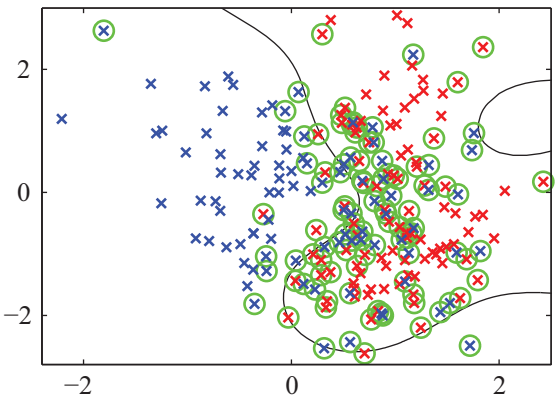


Рис. 7.4. Метод ν-SVM, примененный к неразделимому набору данных в двух измерениях. Опорные векторы обозначаются кружочками

Хотя предсказания для новых входных данных производятся с использованием только опорных векторов, фаза обучения (т.е. определение параметров $\mathbf{a}$ и b) использует все множество данных, и поэтому важно иметь эффективные алгоритмы для решения задачи квадратичного программирования. Прежде всего отметим, что целевая функция $\tilde{L}(\mathbf{a})$, заданная формулами (7.10) или (7.32), является квадратичной, и поэтому любой локальный оптимум также будет глобальным, если ограничения определяют выпуклую область (что они и делают вследствие их линейности). Прямое решение задачи квадратичного программирования с использованием традиционных методов часто неосуществимо из-за высоких требований к скорости вычислений и объему памяти, поэтому необходимо найти более практические подходы. *Метод фрагментации* (Vapnik, 1982) использует тот факт, что значение лагранжиана не изменяется, если мы удалим строки и столбцы матрицы ядра, соответствующие множителям Лагранжа, которые имеют нулевое значение. Это позволяет разбить всю задачу квадратичного программирования на ряд меньших задач, цель которых в конечном итоге определить все ненулевые множители Лагранжа и отбросить остальные. Фрагментацию можно реализовать с использованием *защи-*

щенного метода сопряженных градиентов (Burges, 1998). Хотя по приблизительным оценкам фрагментирование уменьшает размер матрицы в квадратичной функции с количества точек исходных данных в квадрате до количества ненулевых множителей Лагранжа в квадрате, даже это число может быть слишком большим, чтобы поместиться в памяти компьютера для крупномасштабных приложений. *Методы декомпозиции* (Osuna *et al.*, 1996) также сводятся к решению ряда задач квадратичного программирования меньшего размера, но они сконструированы таким образом, что каждая из них имеет фиксированный размер, и поэтому метод может применяться к произвольно большим множествам данных. Тем не менее он по-прежнему связан с численным решением подзадач квадратичного программирования и поэтому может быть проблематичным и дорогостоящим. Один из самых популярных подходов к обучению метода опорных векторов называется *последовательной минимальной оптимизацией*, или SMO (Platt, 1999). Он доводит концепцию фрагментации до предела и рассматривает только два множителя Лагранжа за раз. В этом случае подзадача может быть решена аналитически, тем самым полностью исключая численное решение задачи квадратичного программирования. Выбор пары множителей Лагранжа для рассмотрения на каждом шаге осуществляется по эвристическим правилам. На практике обнаружено, что сложность метода SMO в зависимости от количества точек исходных данных колеблется от линейного до квадратичного в зависимости от конкретного приложения.

Мы видели, что ядра соответствуют скалярным произведениям в пространствах признаков, которые могут иметь большую или даже бесконечную размерность. Поскольку непосредственная работа с ядрами позволяет избежать явного представления пространства признаков, может показаться, что метод опорных векторов каким-то образом снимает “проклятие размерности” (***см. раздел 1.4***). Однако это не так, поскольку существуют ограничения на значения функций, которые ограничивают эффективную размерность пространства признаков. Чтобы увидеть это, рассмотрим простое полиномиальное ядро второго порядка, которое мы можем разложить по его компонентам:

$$\begin{aligned} k(\mathbf{x},\mathbf{z}) &= \left(1+\mathbf{x}^{\mathrm{T}}\mathbf{z}\right)^2 = \left(1+x_1z_1+x_2z_2\right)^2 = \\ &= 1+2x_1z_1+2x_2z_2+x_1^2z_1^2+2x_1z_1x_2z_2+x_2^2z_2^2 = \\ &= \left(1,\sqrt{2}x_1,\sqrt{2}x_2,x_1^2,\sqrt{2}x_1x_2,x_2^2\right)\left(1,\sqrt{2}z_1,\sqrt{2}z_2,z_1^2,\sqrt{2}z_1z_2,z_2^2\right)^{\mathrm{T}} = \\ &= \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{z}). \end{aligned} \tag{7.42}$$

Таким образом, это ядро представляет собой скалярное произведение в пространстве признаков, имеющем шесть измерений, в котором отображение из входного пространства в пространство признаков описывается векторной функцией $\boldsymbol{\phi}(\mathbf{x})$. Однако коэффициенты, взвешивающие эти разные функции, ограничены конкретными формами. Таким образом, любой набор точек в исходном двумерном пространстве $\mathbf{x}$ лежал бы точно на двумерном нелинейном многообразии, вложенном в шестимерное пространство признаков.

Мы уже подчеркивали тот факт, что метод опорных векторов не дает вероятностных результатов, а вместо этого принимает решения о классификации новых входных векторов. Veropoulos *et al.* (1999) предложили модификации метода SVM, позволяющие контролировать компромисс между ложноположительными и ложноотрицательными ошибками. Однако, если мы хотим использовать SVM в качестве модуля в большей вероятностной системе, нам необходимы вероятностные предсказания метки класса t для новых входов $\mathbf{x}$.

Чтобы решить эту проблему, Platt (2000) предложил аппроксимацию логистической сигмоиды на выходах ранее обученного метода опорных векторов. В частности, предполагается, чтобы требуемая условная вероятность имела вид

$$p(t = 1|\mathbf{x}) = \sigma(Ay(\mathbf{x}) + B), \tag{7.43}$$

где функция $y(\mathbf{x})$ определяется формулой (7.1). Значения параметров A и B определяются путем минимизации функции перекрестной энтропии, определенной на обучающем множестве, состоящем из пар значений $y(\mathbf{x}_n)$ и t_n. Чтобы избежать серьезного переобучения, данные, используемые для аппроксимации сигмоиды, должны быть независимыми от данных, используемых для обучения оригинального метода SVM. Этот двухэтапный подход эквивалентен предположению, что выходной сигнал $y(\mathbf{x})$ метода опорных векторов представляет собой логарифм отношения шансов того, что вектор $\mathbf{x}$ принадлежит классу $t = 1$. Поскольку процедура обучения SVM не предназначена для этого специально, этот метод может дать плохую аппроксимацию апостериорных вероятностей (Tipping, 2001).

7.1.2. Связь с логистической регрессией

Как и в случае разделимого случая, мы можем повторно преобразовать метод SVM для неразделимых распределений в терминах минимизации регуляризованной функции ошибок. Это также позволит выделить сходства и различия с моделью логистической регрессии (***см. раздел 4.3.2***).

Мы видели, что для точек исходных данных, которые находятся на правильной стороне от границы зазора и, следовательно, удовлетворяют условию $y_n t_n \geq 1$, име-

ем $\xi_n = 0$, а для остальных точек $\xi_n = 1 - y_n t_n$. Таким образом, целевая функция (7.21) может быть записана (с точностью до общей мультипликативной константы) в виде

$$\sum_{n=1}^{N} E_{\mathrm{SV}}\left(y_n t_n\right) + \lambda \|\mathbf{w}\|^2, \tag{7.44}$$

где $\lambda = (2C)^{-1}$ и $\mathrm{E}_{SV}(\cdot)$ — *кусочно-линейная функция ошибок* (hinge error), определяемая формулой

$$E_{\mathrm{SV}}(y_n t_n) = [1 - y_n t_n]_+, \tag{7.45}$$

где $[\cdot]_+$ обозначает положительную часть. Кусочно-линейная функция ошибок, называемая так из-за ее формы, показана на рис. 7.5. Ее можно рассматривать как аппроксимацию ошибки классификации, т.е. функцию ошибок, которую мы хотели бы свести к минимуму, что также показано на рис. 7.5.

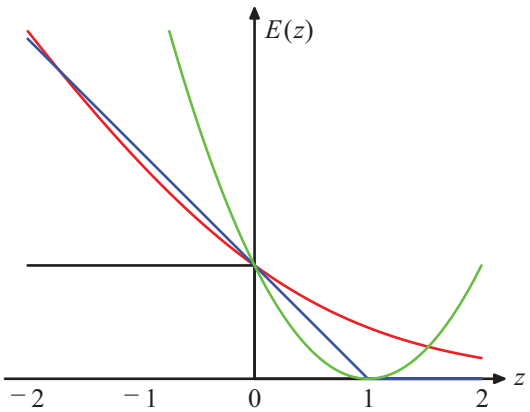


Рис. 7.5. График кусочно-линейной функции ошибок, используемой в методе опорных векторов (синяя кривая), и функция ошибок для логистической регрессии (красная кривая), масштабированная с коэффициентом 1/1n 2, чтобы она проходила через точку (0, 1). Черным цветом показаны ошибки классификации, зеленым — квадратическая ошибка

Когда мы рассматривали модель логистической регрессии в ***разделе 4.3.2***, нам было удобно работать с целевой переменной $t \in \{0, 1\}$. Для сравнения с методом опорных векторов мы сначала переформулируем логическую регрессию максимального правдоподобия с использованием целевой переменной $t \in \{-1, 1\}$. Для этого заметим, что $p(t = 1|y) = \sigma(y)$, где $y(\mathbf{x})$ задается формулой (7.1), а $\sigma(y)$ — логистическая сигмоида, определяемая формулой (4.59). Отсюда следует, что

$p(t = -1|y) = 1 - \sigma(y) = \sigma(-y)$, где мы использовали свойства логистической сигмоиды, и поэтому можем записать:

$$p(t|y) = \sigma(yt). \tag{7.46}$$

Отсюда можно построить функцию ошибок, взяв отрицательный логарифм функции правдоподобия, которая с квадратичным регуляризатором примет вид (***см. упражнение 7.6***).

$$\sum_{n=1}^{N} E_{\mathrm{LR}}\left(y_n t_n\right) + \lambda \left\|\mathbf{w}\right\|^2, \tag{7.47}$$

где

$$E_{\mathrm{LR}}(yt) = \ln(1 + \exp(-yt)). \tag{7.48}$$

Для сравнения с другими функциями ошибок мы можем разделить эту формулу на ln(2) так, чтобы функция ошибок проходила через точку (0, 1). Эта функция с измененной ошибкой также показана на рис. 7.5 и имеет аналогичную форму для функции ошибок метода опорных векторов. Основное различие заключается в том, что плоская область в $\mathrm{E}_{\mathrm{SV}}(yt)$ приводит к разреженным решениям.

Как логистическая ошибка, так и кусочно-линейная функция потерь могут рассматриваться как непрерывные приближения к уровню ошибочной классификации. Другая функция непрерывной ошибки, которая иногда использовалась для решения задач классификации, представляет собой квадратичную ошибку, которая также показана на рис. 7.5. Однако оно приписывает большой вес точкам, которые были классифицированы правильно, но расположены далеко от границы решения на правильной стороне. Такие точки будут сильно перевешивать вклад ошибочно классифицированных точек, и поэтому, если целью является минимизация уровня ошибочной классификации, лучшим выбором будет монотонно убывающая функция ошибок.

7.1.3. Многоклассовые варианты SVM

Метод опорных векторов в основном применяется для решения задачи бинарной классификации. На практике, однако, часто приходится решать задачи, связанные с $K > 2$ классами. Поэтому были предложены различные методы для объединения нескольких двухклассовых SVM для создания многоклассового метода классификации.

Одним из широко используемых подходов (Vapnik, 1998) является построение K отдельных методов SVM, в которых k-я модель $y_k(\mathbf{x})$ обучается с использованием данных из класса $\mathcal{C}_k$ в качестве положительных примеров и данных из оставшихся K–1 классов — как отрицательных примеров. Этот подход называ-

ется "*один против остальных*". Однако на рис. 4.2 показано, что использование решений отдельных методов классификации может привести к несогласованным результатам, при которых вектор назначается нескольким классам одновременно. Иногда эту проблему можно решить, делая прогнозы для новых входных данных $\mathbf{x}$ по правилу

$$y(\mathbf{x}) = \max_k y_k(\mathbf{x}). \tag{7.49}$$

К сожалению, этот эвристический подход имеет недостаток: разные методы классификации обучаются на разных задачах, и нет гарантии, что реальные величины $y_k(\mathbf{x})$ для разных методов классификации будут иметь соответствующие масштабы.

Еще одна проблема с подходом "один против остальных" заключается в том, что обучающие множества не сбалансированы. Например, если у нас есть десять классов с одинаковым количеством обучающих точек, то отдельные методы классификации обучаются на множествах данных, содержащих 90% отрицательных примеров и только 10% положительных, и симметрия исходной задачи теряется. Один из вариантов схемы "один против остальных" был предложен Lee *et al.* (2001). В этом методе целевые значения изменяются так, чтобы положительный класс имел целевое значение +1, а отрицательному соответствовало значение $-1/(K-1)$.

Weston and Watkins (1999) определяют одну целевую функцию для обучения всех K методов SVM одновременно, основываясь на максимизации зазора между каждым классом и остальными классами. Однако это может привести к значительному замедлению обучения, поскольку, вместо того чтобы решать K отдельных задач оптимизации по N точкам с общей сложностью $O(KN^2)$, необходимо решить одну задачу оптимизации размера $(K-1)N$ с общей сложностью $O(K^2N^2)$.

Другой подход состоит в том, чтобы обучить $K(K-1)/2$ различных бинарных вариантов SVM для всех возможных пар классов, а затем классифицировать тестовые точки в соответствии с наибольшим количеством голосов. Этот подход иногда называют "*каждый против каждого*". Мы уже видели на рис. 4.2, что это может привести к двусмысленности в полученной классификации. Кроме того, для больших K этот подход требует значительно большего времени обучения, чем подход "один против остальных". Аналогично для оценки тестовых точек требуется значительно больше вычислений.

Последнюю задачу можно решить путем организации бинарных классификаторов в виде направленного ациклического графа (не путайте его с вероятност-

ной графовой моделью), что приводит к методу DAGSVM (Platt *et al.*, 2000). Для K классов метод DAGSVM имеет в общей сложности $K(K-1)/2$ классификаторов, и для классификации новой тестовой точки требуется вычислить только $K-1$ бинарных классификаций с использованием конкретных классификаторов в зависимости от пути в графе.

Другой подход к многоклассовой классификации основан на кодах с исправлением ошибок. Он был разработан Dietterich and Bakiri (1995) и применен для метода опорных векторов в работе Allwein *et al.* (2000). Его можно рассматривать как обобщение схемы голосования "каждый против каждого", в которой для подготовки отдельных классификаторов используются более общие разделения классов. Сами K классов представляются в виде отдельных наборов ответов выбранных бинарных классификаторов. Вместе с подходящей схемой декодирования это обеспечивает устойчивость к ошибкам и неоднозначности в выводах отдельных классификаторов. Хотя применение метода SVM к задачам классификации многих классов остается открытой проблемой, на практике подход "один против остальных" используется наиболее широко, несмотря на его специфическую формулировку и практические ограничения.

Существуют также *одноклассовые методы опорных векторов*, которые решают задачу обучения без учителя, связанную с оценкой плотности вероятности. Однако вместо моделирования плотности данных эти методы направлены на то, чтобы найти гладкую границу, охватывающую область высокой плотности. Граница выбирается так, чтобы представлять квантиль плотности, т.е. вероятность того, что точка данных, полученная из распределения, попадет в эту область, задается фиксированным числом от 0 до 1, которое задано заранее. Это более ограниченная задача, чем оценка полной плотности, но ее может быть достаточно для конкретных приложений. Предложены два подхода к этой задаче с использованием метода опорных векторов. Алгоритм Schölkopf *et al.* (2001) пытается найти гиперплоскость, которая отделяет все, кроме фиксированной доли ν обучающих данных от начала координат, и в то же время максимизирует расстояние (зазор) гиперплоскости от начала координат, а Tax and Duin (1999) строят наименьшую сферу в пространстве признаков, содержащую все, кроме доли ν точек исходных данных. Для ядер $k(\mathbf{x}, \mathbf{x}')$, которые являются функциями только $\mathbf{x}-\mathbf{x}'$, оба алгоритма эквивалентны.

7.1.4. Метод SVM для регрессии

Теперь распространим метод опорных векторов на задачи регрессии и в то же время сохраним свойство разреженности (***см. раздел 3.1.4***). В простой линей-

ной регрессии мы минимизируем регуляризованную функцию ошибок, заданную формулой

$$\frac{1}{2}\sum_{n=1}^{N}\{y_n - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2. \tag{7.50}$$

Для получения разреженных решений функция квадратичных ошибок заменяется *ε-нечувствительной функцией ошибок* (Vapnik, 1995), которая равна нулю, если абсолютная разница между прогнозом $y(\mathbf{x})$ и целевым значением t меньше, чем ε, где $\varepsilon > 0$. Простой пример ε-нечувствительной функции ошибок, имеющей линейный штраф за ошибки вне области нечувствительности, определяется формулой

$$E_\varepsilon\left(y(\mathbf{x}) - t\right) = \begin{cases} 0, \text{ если } \left|y(\mathbf{x}) - t\right| < \varepsilon, \\ \left|y(\mathbf{x}) - t\right| - \varepsilon \text{ в противном случае.} \end{cases} \tag{7.51}$$

Она показана на рис. 7.6.

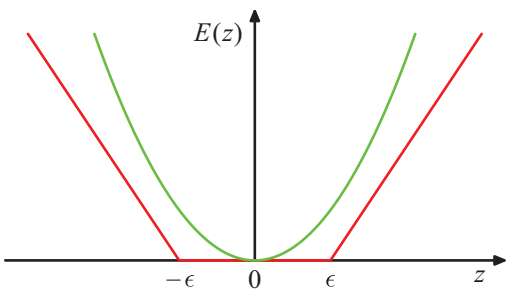


Рис. 7.6. График ε-нечувствительной функции ошибок (красный цвет), при которой ошибка линейно возрастает с расстоянием за пределами области нечувствительности. Для сравнения также показана функция квадратичной ошибки (зеленый цвет)

Следовательно, мы должны минимизировать регуляризованную функцию ошибок, заданную формулой

$$C\sum_{n=1}^{N} E_\varepsilon\left(y(\mathbf{x}_n) - t_n\right) + \frac{1}{2}\|\mathbf{w}\|^2, \tag{7.52}$$

где $y(\mathbf{x})$ задается формулой (7.1). По соглашению (обратный) параметр регуляризации, обозначаемый C, стоит перед ошибкой.

Как и ранее, мы можем переформулировать задачу оптимизации, введя фиктивные переменные. Для каждой точки данных $\mathbf{x}_n$ нам понадобятся две фиктивные переменные, $\xi_n \geq 0$ и $\hat{\xi}_n \geq 0$, где $\xi_n > 0$ соответствует точке, для которой

$t_n > y(\mathbf{x}_n) + \varepsilon$, а $\hat{\xi}_n > 0$ соответствует точке, для которой $t_n < y(\mathbf{x}_n) - \varepsilon$ (рис. 7.7). Условие для целевой точки, лежащей внутри ε-трубки, состоит в том, что $y_n - \varepsilon \le$ $\le t_n \le y_n + \varepsilon$, где $y_n = y(\mathbf{x}_n)$. Фиктивные переменные позволяют точкам находиться снаружи трубки, если фиктивные переменные отличны от нуля и выполняются соответствующие условия:

$$t_n \le y(\mathbf{x}_n) + \varepsilon + \xi_n, \tag{7.53}$$

$$t_n \ge y(\mathbf{x}_n) - \varepsilon - \hat{\xi}_n. \tag{7.54}$$

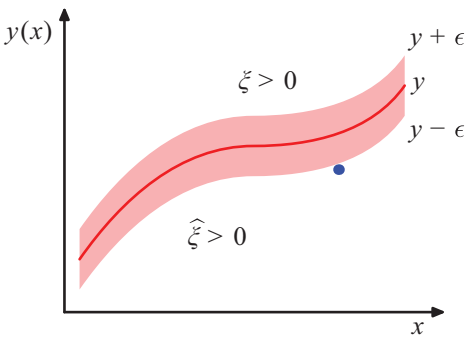


Рис. 7.7. Применение метода SVM для регрессии. Показаны регрессионная кривая вместе с ε-нечувствительной трубкой, а также примеры фиктивных переменных ξ и $\hat{\xi}_n$. Точки над ε-трубкой удовлетворяют условию $\xi > 0$ и $\hat{\xi}_n = 0$, точки ниже ε-трубки удовлетворяют условию $\xi = 0$ и $\hat{\xi}_n > 0$, а точки внутри ε-трубки удовлетворяют условию $\xi = \hat{\xi}_n = 0$

Функция ошибок для регрессии по методу опорных векторов может быть записана как

$$C\sum_{n=1}^{N}\left(\xi_n + \hat{\xi}_n\right) + \frac{1}{2}\|\mathbf{w}\|^2. \tag{7.55}$$

Данную функцию необходимо минимизировать при ограничениях $\xi_n \ge 0$ и $\hat{\xi}_n \ge 0$, а также (7.53) и (7.54). Этого можно добиться, введя множители Лагранжа $a_n \ge 0$, $\hat{a}_n \ge 0$, $\mu_n \ge 0$ и $\hat{\mu}_n \ge 0$ и оптимизируя лагранжиан:

$$\begin{aligned} L = C\sum_{n=1}^{N}\left(\xi_n + \hat{\xi}_n\right) + \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N}\left(\mu_n\xi_n + \hat{\mu}_n\hat{\xi}_n\right) - \\ -\sum_{n=1}^{N}a_n\left(\varepsilon + \xi_n + y_n - t_n\right) - \sum_{n=1}^{N}\hat{a}_n\left(\varepsilon + \hat{\xi}_n - y_n + t_n\right). \end{aligned} \tag{7.56}$$

Теперь заменим $y(\mathbf{x})$ на (7.1), а затем приравняем производные от лагранжиана по $\mathbf{w}$, b, ξ_n и $\hat{\xi}_n$ к нулю, получая

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{n=1}^{N} (a_n - \hat{a}_n) \boldsymbol{\phi}(\mathbf{x}_n), \tag{7.57}$$

$$\frac{\partial L}{\partial b} = 0 \quad \Rightarrow \quad \sum_{n=1}^{N} (a_n - \hat{a}_n) = 0, \tag{7.58}$$

$$\frac{\partial L}{\partial \xi_n} = 0 \quad \Rightarrow \quad a_n + \mu_n = C, \tag{7.59}$$

$$\frac{\partial L}{\partial \hat{\xi}_n} = 0 \quad \Rightarrow \quad \hat{a}_n + \hat{\mu}_n = C. \tag{7.60}$$

Используя эти результаты для исключения соответствующих переменных из лагранжиана, мы видим, что двойственная задача сводится к максимизации функции (***см. упражнение 7.7***):

$$\begin{aligned} \tilde{L}(\mathbf{a}, \hat{\mathbf{a}}) = &-\frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} (a_n - \hat{a}_n)(a_m - \hat{a}_m) k(\mathbf{x}_n, \mathbf{x}_m) - \\ &-\varepsilon \sum_{n=1}^{N} (a_n + \hat{a}_n) + \sum_{n=1}^{N} (a_n - \hat{a}_n) t_n \end{aligned} \tag{7.61}$$

относительно $\{a_n\}$ и $\{\hat{a}_n\}$, где мы ввели ядро $k(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}')$. Как и в предыдущем случае, это задача максимизации с ограничениями, и чтобы найти ограничения, заметим, что должны выполняться условия $a_n > 0$ и $\hat{a}_n \geq 0$, потому что это — множители Лагранжа. Кроме того, условия $\mu_n > 0$ и $\hat{\mu}_n \geq 0$ вместе с (7.59) и (7.60) означают, что $a_n \leq C$ и $\hat{a}_n \leq C$, и поэтому снова получаем квадратные ограничения:

$$0 \leq a_n \leq C, \tag{7.62}$$

$$0 \leq \hat{a}_n \leq C \tag{7.63}$$

вместе с условием (7.58).

Подставляя (7.57) в (7.1), мы видим, что прогнозы для новых входных данных могут быть сделаны с использованием формулы

$$y(\mathbf{x}) = \sum_{n=1}^{N} (a_n - \hat{a}_n) k(\mathbf{x}, \mathbf{x}_n) + b, \tag{7.64}$$

которая снова выражается через ядро.

Соответствующие условия Каруша–Куна–Таккера (ККТ), которые утверждают, что произведение двойственных переменных и ограничений в решении должно обращаться в нуль, задаются формулами

$$a_n(\varepsilon + \xi_n + y_n - t_n) = 0, \tag{7.65}$$

$$\hat{a}_n\left(\varepsilon + \hat{\xi}_n - y_n + t_n\right) = 0, \tag{7.66}$$

$$(C - a_n)\xi_n = 0, \tag{7.67}$$

$$\left(C - \hat{a}_n\right)\hat{\xi}_n = 0. \tag{7.68}$$

Из них мы можем получить несколько полезных результатов. Прежде всего отметим, что коэффициент a_n может быть отличным от нуля, если $\varepsilon + \xi_n + y_n - t_n = 0$, откуда следует, что обучающая точка либо лежит на верхней границе ε-трубки ($\xi_n = 0$), либо над верхней границей ($\xi_n > 0$). Точно так же ненулевое значение для $\hat{a}_n$ подразумевает, что $\varepsilon + \hat{\xi}_n - y_n + t_n = 0$ и такие точки должны лежать либо на нижней границе трубки, либо ниже нее.

Кроме того, два ограничения, $\varepsilon + \xi_n + y_n - t_n = 0$ и $\varepsilon + \hat{\xi}_n - y_n + t_n$ несовместимы, что легко увидеть, сложив их вместе и отметив, что ξ_n и $\hat{\xi}_n$ не отрицательны, в то время как ε — строго положительное число, поэтому для каждой точки исходных данных $\mathbf{x}_n$ либо a_n, либо $\hat{a}_n$ (или обе переменные) должны быть равными нулю.

Опорными векторами являются те точки исходных данных, которые вносят вклад в прогнозы, заданные формулой (7.64), иначе говоря, те, для которых либо $a_n \neq 0$, либо $\hat{a}_n \neq 0$. Это точки, которые лежат на границе трубки или вне трубки. Для всех точек внутри трубки выполняются условия $a_n = \hat{a}_n = 0$. Мы снова получаем разреженное решение, и единственными членами, которые должны быть вычислены в прогностической модели (7.64), являются те, которые включают в себя опорные векторы.

Параметр b можно найти, рассматривая точку исходных данных, для которой $0 < a_n < C$, которая с учетом (7.67) должна удовлетворять условию $\xi_n = 0$, и поэтому с учетом (7.65) удовлетворять условию $\varepsilon + y_n - t_n = 0$. Используя (7.1) и решая уравнение относительно b, получим:

$$\begin{aligned} b &= t_n - \varepsilon - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}\left(\mathbf{x}_n\right) = \\ &= t_n - \varepsilon - \sum_{m=1}^{N}\left(a_m - \hat{a}_m\right)k\left(\mathbf{x}_n, \mathbf{x}_m\right), \end{aligned} \tag{7.69}$$

где мы использовали условие (7.57). Мы можем получить аналогичный результат, рассматривая точку, для которой $0 < \hat{a}_n < C$. На практике лучше усреднять по всем таким оценкам b.

Как и в случае классификации, существует альтернативная формулировка SVM для регрессии, в которой параметр, определяющий сложность, имеет более интуитивную интерпретацию (Schölkopf *et al.*, 2000). В частности, вместо того, чтобы фиксировать ширину ε-нечувствительной области, мы фиксируем параметр ν, который ограничивает долю точек, лежащих вне трубки. Это предполагает максимизацию функции

$$\tilde{L}(\mathbf{a}, \hat{\mathbf{a}}) = -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}(a_n - \hat{a}_n)(a_m - \hat{a}_m)k(\mathbf{x}_n, \mathbf{x}_m) + \\ + \sum_{n=1}^{N}(a_n - \hat{a}_n)t_n \tag{7.70}$$

с учетом ограничений

$$0 \leq a_n \leq C/N, \tag{7.71}$$

$$0 \leq \hat{a}_n \leq C/N, \tag{7.72}$$

$$\sum_{n=1}^{N}(a_n - \hat{a}_n) = 0, \tag{7.73}$$

$$\sum_{n=1}^{N}(a_n + \hat{a}_n) \leq \nu C. \tag{7.74}$$

Можно показать, что за пределы нечувствительной трубки выходят не более чем νN точек исходных данных, в то время как по крайней мере νN точек исходных данных являются опорными векторами и поэтому лежат либо на трубке, либо снаружи.

Использование метода опорных векторов для решения задачи регрессии иллюстрируется синусоидальным множеством данных, показанным на рис. 7.8 (***см. приложение А***). Здесь параметры ν и C выбраны вручную. На практике их значения обычно определяются с помощью перекрестной проверки.

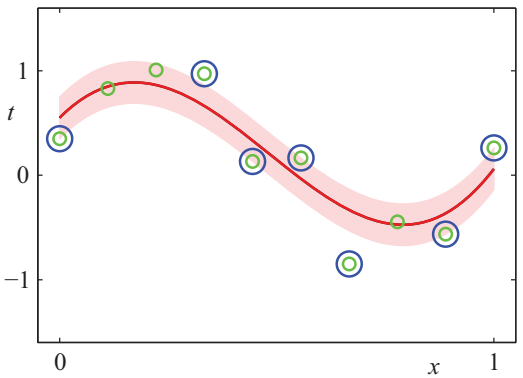


Рис. 7.8. Применение метода ν-SVM для задачи регрессии с искусственным синусоидальным набором данных и гауссовых ядер. Кривая прогнозируемой регрессии показана красной кривой, а ε-нечувствительная трубка соответствует заштрихованной области. Кроме того, точки исходных данных показаны зеленым цветом, а опорные векторы — синими кружочками

7.1.5. Теория вычислительного обучения

Исторически метод опорных векторов обычно анализировался и обосновывался с использованием теоретического подхода, известного как *теория вычислительного обучения*, также иногда называемого *теорией статистического обучения* (Anthony and Biggs, 1992; Kearns and Vazirani, 1994; Vapnik, 1995; Vapnik, 1998). Этот термин впервые ввел Valiant (1984), который сформулировал теорию *приближенно корректного обучения с высокой вероятностью*, или PAC (probably approximately correct). Цель теории PAC состоит в том, чтобы понять, насколько большим должно быть множество исходных данных, чтобы обеспечить хорошее обобщение. Она также дает оценки вычислительной стоимости обучения, хотя мы их здесь не рассматриваем.

Предположим, множество данных $\mathcal{D}$ размера N извлечено из генеральной совокупности с совместным распределением $p(\mathbf{x}, \mathbf{t})$, где $\mathbf{x}$ — входная переменная, а $\mathbf{t}$ — метка класса. Ограничимся ситуацией, в которой шум отсутствует, а метки классов определяются некоторой (неизвестной) детерминированной функцией $\mathbf{t} = g(\mathbf{x})$. Придерживаясь теории PAC, мы говорим, что функция $\mathbf{f}(\mathbf{x}; \mathcal{D})$, извлеченная из пространства $\mathcal{F}$ таких функций на основе обучающего множества $\mathcal{D}$, имеет хорошее обобщение, если ожидаемая частота ошибок не превышает некоторый заранее заданный порог ε, т.е.

$$E_{\mathbf{x},\mathbf{t}}\left[I\left(\mathbf{f}\left(\mathbf{x};\mathcal{D}\right)\neq\mathbf{t}\right)\right]<\varepsilon, \tag{7.75}$$

где $I(\cdot)$ — индикаторная функция, а математическое ожидание вычисляется относительно распределения $p(\mathbf{x}, \mathbf{t})$. Величина в левой части является случайной, так как зависит от обучающего множества $\mathcal{D}$, а для теории PAC требуется, чтобы для множества данных $\mathcal{D}$, случайным образом извлеченного из генеральной совокупности с распределением $p(\mathbf{x}, \mathbf{t})$, неравенство (7.75) выполнялось с вероятностью, большей, чем $1–\delta$. Здесь δ — еще один заранее заданный параметр, а термин "приближенно корректный с высокой вероятностью" отражает требование, чтобы с большой вероятностью (больше $1 – \delta$) частота ошибок была небольшой (меньше ε). При заданном выборе пространства моделей $\mathcal{F}$ и заданных параметрах ε и δ обучение PAC направлено на обеспечение границ минимального размера N множества данных, необходимого для соответствия этому критерию. Ключевым количеством в теории PAC является *размерность Вапника–Червоненкиса,* или VC, которая представляет собой меру сложности пространства функций и позволяет распространять подход PAC на пространства, содержащие бесконечное количество функций.

Оценки, полученные в рамках теории PAC, часто описывают наихудший случай, поскольку они относятся к *произвольному* выбору распределения $p(\mathbf{x}, \mathbf{t})$, если обучающие и тестовые выборки независимо извлекаются из одной и той же генеральной совокупности, и к *произвольному* выбору функции $\mathbf{f}(\mathbf{x})$, принадлежащей $\mathcal{F}$. В реальных приложениях машинного обучения мы имеем дело с распределениями, которые имеют значительную регулярность, например, когда большие области исходного пространства имеют одну и ту же метку класса. Вследствие отсутствия каких-либо предположений о форме распределения границы PAC очень консервативны, иначе говоря, они сильно переоценивают размер множеств данных, необходимых для достижения заданной точности обобщения. По этой причине оценки PAC почти не нашли практических приложений.

Одной из попыток улучшить точность границ PAC является *PAC-байесовский подход* (McAllester, 2003), который рассматривает распределение по пространству $\mathcal{F}$ функций, напоминающее априорное распределение в байесовском подходе. Он все еще рассматривает произвольный выбор $p(\mathbf{x}, \mathbf{t})$, и поэтому, хотя границы более узкие, они по-прежнему очень консервативны.

7.2. Метод релевантных векторов

Метод опорных векторов использовался в различных приложениях для классификации и регрессии. Тем не менее он страдает от ряда ограничений, некоторые из которых уже описаны в этой главе. В частности, результаты SVM представляют собой решения, а не апостериорные вероятности. Кроме того, SVM из-

начально был разработан для двух классов, и его расширение на случай $K > 2$ классов проблематично. Существует параметр сложности C, или ν (а также параметр ε в случае регрессии), который должен быть найден с помощью контроля на отложенных данных, например перекрестной проверки. Наконец, прогнозы выражаются в виде линейных комбинаций ядер, которые центрированы на точках обучающих данных и должны быть положительно определенными.

Метод релевантных векторов, или RVM (Tipping, 2001), — это байесовский разреженный ядерный метод для регрессии и классификации, который обладает многими преимуществами SVM и не имеет его основных ограничений. Кроме того, обычно он приводит к появлению гораздо более разреженных моделей, что способствует более быстрой работе с тестовыми данными при сохранении сопоставимой ошибки обобщения.

В отличие от метода SVM, нам будет удобнее сначала ввести регрессионную форму RVM, а затем рассмотреть вопрос о его распространении на задачи классификации.

7.2.1. Метод RVM для регрессии

Метод релевантных векторов для регрессии является линейной моделью, уже изученной в главе 3, но с модифицированным априорным распределением, порождающим разреженные решения. Эта модель определяет условное распределение для действительной целевой переменной t при заданном входном векторе $\mathbf{x}$, которое принимает вид

$$p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}), \beta^{-1}), \tag{7.76}$$

где $\beta = \sigma^{-2}$ — точность шума (обратная дисперсия шума), а математическое ожидание задается линейной моделью вида

$$y(\mathbf{x}) = \sum_{i=1}^{M} w_i \phi_i(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}) \tag{7.77}$$

с фиксированными нелинейными базисными функциями $\phi_i(\mathbf{x})$, которые обычно включают постоянный член, так что соответствующий весовой параметр представляет собой смещение.

Метод релевантных векторов является конкретным примером модели, предназначенной для отражения структуры метода опорных векторов. В частности, базисные функции задаются ядрами, причем каждое ядро связано с каждой из точек обучающих данных. Тогда общее выражение (7.77) принимает SVM-подобную форму:

$$y(\mathbf{x}) = \sum_{n=1}^{N} w_n k(\mathbf{x}, \mathbf{x}_n) + b, \tag{7.78}$$

где b — параметр смещения. Количество параметров в этом случае равно $M = N + 1$, а $y(\mathbf{x})$ имеет тот же вид, что и прогностическая модель (7.64) для SVM, за исключением того, что коэффициенты a_n здесь обозначаются как w_n. Следует подчеркнуть, что последующий анализ справедлив для произвольного выбора базисной функции, а для общности будем работать с формой (7.77). В отличие от SVM, в методе RVM нет никаких ограничений на положительную определенность ядра, а базисные функции не привязаны ни к количеству, ни по местоположению обучающих точек данных.

Предположим, нам дано множество, состоящее из N наблюдений входного вектора $\mathbf{x}$, которое мы обозначим матрицей данных $\mathbf{X}$, n-я строка которой представляет собой вектор $\mathbf{x}_n^T$, где $n = 1, ..., N$. Соответствующие целевые значения задаются вектором $\mathbf{t} = (t_1, ..., t_N)^{\mathrm{T}}$. Таким образом, функция правдоподобия определяется формулой

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^{N} p(t_n|\mathbf{x}_n, \mathbf{w}, \beta). \tag{7.79}$$

Затем мы вводим априорное распределение по вектору параметров $\mathbf{w}$ и, как в главе 3, рассматриваем априорное нормальное распределение с нулевым математическим ожиданием. Однако ключевым отличием метода RVM является то, что мы вводим отдельный гиперпараметр α_i для каждого из весовых параметров w_i вместо одного общего гиперпараметра. Таким образом, априорное распределение весов принимает вид

$$p(\mathbf{w}|\boldsymbol{\alpha}) = \prod_{i=1}^{M} \mathcal{N}(w_i|0, \alpha_i^{-1}), \tag{7.80}$$

где α_i представляет собой точность соответствующего параметра w_i; $\boldsymbol{\alpha}$ обозначает вектор $(\alpha_1, ..., \alpha_M)^{\mathrm{T}}$. В дальнейшем мы увидим, что при максимизации обоснованности этих гиперпараметров значительная их часть стремится к бесконечности, а соответствующие весовые параметры имеют апостериорные распределения, сосредоточенные в нуле. Таким образом, базисные функции, связанные с этими параметрами, не играют никакой роли в прогнозах, сделанных моделью, и поэтому эффективно сокращаются, что приводит к разреженной модели.

Используя результат (3.49) для моделей линейной регрессии, мы видим, что апостериорное распределение весов снова является нормальным и принимает вид

$$p(\mathbf{w}|\mathbf{t}, \mathbf{X}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \mathcal{N}(\mathbf{w}|\mathbf{m}, \boldsymbol{\Sigma}), \tag{7.81}$$

где математическое ожидание и ковариационная матрица задаются выражениями

$$\mathbf{m} = \beta \boldsymbol{\Sigma} \boldsymbol{\Phi}^{\mathrm{T}} \mathbf{t}, \tag{7.82}$$

$$\boldsymbol{\Sigma} = (\mathbf{A} + \beta \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi})^{-1}, \tag{7.83}$$

где $\boldsymbol{\Phi}$ — матрица плана с $N \times M$ с элементами $\Phi_{ni} = \phi_i(\mathbf{x}_n)$ для $i = 1,..., N$, $\Phi_{nM} = 1$ для $n = 1, ..., N$ и $\mathbf{A} = \mathrm{diag}(\alpha_i)$.

Значения $\boldsymbol{\alpha}$ и $\boldsymbol{\beta}$ определяются с использованием метода максимального правдоподобия второго типа (***см. раздел 3.5***), известного как *аппроксимация обоснованности*, в котором максимизируется маргинальное правдоподобие, полученное путем интегрирования по весовым параметрам:

$$p\left(\mathbf{t}\middle|\mathbf{X}, \boldsymbol{\alpha}, \beta\right) = \int p\left(\mathbf{t}\middle|\mathbf{X}, \mathbf{w}, \beta\right) p\left(\mathbf{w}\middle|\boldsymbol{\alpha}\right) d\mathbf{w}. \tag{7.84}$$

Поскольку это выражение представляет собой свертку двух нормальных распределений, легко вычислить, что логарифмическая функция маргинального правдоподобия имеет вид (***см. упражнение 7.10***)

$$\begin{aligned} p\left(\mathbf{t}\middle|\mathbf{X}, \boldsymbol{\alpha}, \beta\right) &= \ln \mathcal{N}\left(\mathbf{t}\middle|\mathbf{0}, \mathbf{C}\right) = \\ &= -\frac{1}{2}\left\{N \ln(2\pi) + \ln|\mathbf{C}| + \mathbf{t}^{\mathrm{T}} \mathbf{C}^{-1} \mathbf{t}\right\}, \end{aligned} \tag{7.85}$$

где $\mathbf{t} = (t_1, ..., t_N)^{\mathrm{T}}$, и мы определили матрицу $\mathbf{C}$ размера $N \times N$, заданную формулой

$$\mathbf{C} = \beta^{-1}\mathbf{I} + \boldsymbol{\Phi} \mathbf{A}^{-1} \boldsymbol{\Phi}^{\mathrm{T}}. \tag{7.86}$$

Наша цель — максимизировать (7.85) по гиперпараметрам $\boldsymbol{\alpha}$ и β. Это требует лишь небольшой модификации результатов, полученных в ***разделе 3.5*** для аппроксимации обоснованности модели линейной регрессии. Как и прежде, можем определить два подхода. В первом мы просто приравниваем искомые производные маргинального правдоподобия к нулю и получаем следующие уравнения для итерационного приближения (***см. упражнение 7.12***):

$$\alpha_i^{new} = \frac{\gamma_i}{m_i^2}, \tag{7.87}$$

$$\left(\beta^{new}\right)^{-1} = \frac{\left\|\mathbf{t} - \boldsymbol{\Phi}\mathbf{m}\right\|^2}{N - \sum_i \gamma_i}, \tag{7.88}$$

где m_i — i-й компонент апостериорного математического ожидания $\mathbf{m}$, определяемая формулой (7.82). Величина γ_i измеряет, насколько хорошо соответствующий параметр w_i определяется данными и задается формулой (***см. раздел 3.5.3***)

$$\gamma_i = 1 - \alpha_i \Sigma_{ii}, \tag{7.89}$$

в которой Σ_{ii} — i-я диагональный компонент апостериорной ковариационной матрицы $\boldsymbol{\Sigma}$, заданной формулой (7.83). Таким образом, обучение происходит путем выбора начальных значений для $\boldsymbol{\alpha}$ и β, вычисления математического ожидания и ковариационной матрицы апостериорного распределения (7.82) и (7.83) соответственно, а затем итеративного уточнения гиперпараметров по формулам (7.87) и (7.88) и апостериорного математического ожидания и ковариационной матрицы по формулам (7.82) и (7.83), до тех пор, пока не будет выполнен критерий сходимости.

Второй подход заключается в использовании EM-алгоритма, который обсуждается в ***разделе 9.3.4***. Эти два подхода к определению значений гиперпараметров, максимизирующих обоснованность, формально эквивалентны (***см. упражнение 9.23***). Однако с помощью вычислений было показано, что подход, основанный на прямой оптимизации по формулам (7.87) и (7.88), обеспечивает несколько более высокую сходимость (Tipping, 2001).

В результате оптимизации мы обнаруживаем, что доля гиперпараметров $\{\alpha_i\}$ стремится к бо́льшим (в принципе бесконечным) значениям, поэтому весовые параметры w_i, соответствующие этим гиперпараметрам, имеют апостериорные распределения с нулевым математическим ожиданием и дисперсией (***см. раздел 7.2.2***). Таким образом, эти параметры и соответствующие базисные функции $\phi_i(\mathbf{x})$ исключаются из модели и не играют никакой роли в прогнозировании новых данных. В случае моделей вида (7.78) входные данные $\mathbf{x}_n$, соответствующие остальным ненулевым весам, называются *релевантными векторами*, поскольку они идентифицируются с помощью механизма автоматического определения релевантности и аналогичны опорным векторам SVM. Следует, однако, подчеркнуть, что этот механизм достижения разреженности в вероятностных моделях посредством автоматического определения релевантности является довольно универсальным и может применяться к любой модели, выраженной как адаптивная линейная комбинация базисных функций.

Найдя значения $\boldsymbol{\alpha}^*$ и β^* для гиперпараметров, которые максимизируют маргинальное правдоподобие, можно вычислить прогностическое распределение по t для нового входного вектора $\mathbf{x}$. Из (7.76) и (7.81) следует, что это распределение задается формулой (***см. упражнение 7.14***)

$$p\left(t|\mathbf{x},\mathbf{X},\mathbf{t},\boldsymbol{\alpha}^*,\beta^*\right)=\int p\left(t|\mathbf{x},\mathbf{w},\beta^*\right)p\left(\mathbf{w}|\mathbf{X},\mathbf{t},\boldsymbol{\alpha}^*,\beta^*\right)d\mathbf{w}= \\ =\mathcal{N}\left(t|\mathbf{m}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}),\sigma^2(\mathbf{x})\right). \tag{7.90}$$

Таким образом, прогностическое математическое ожидание задается выражением (7.76), где $\mathbf{w}$ устанавливается равным апостериорному математическому ожиданию $\mathbf{m}$, а дисперсия прогностического распределения определяется выражением

$$\sigma^2(\mathbf{x})=(\beta^*)^{-1}+\boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{\phi}(\mathbf{x}), \tag{7.91}$$

где матрица $\boldsymbol{\Sigma}$ задается формулой (7.83), в которой $\boldsymbol{\alpha}$ и β принимают оптимальные значения $\boldsymbol{\alpha}^*$ и β^*. Это знакомый результат (3.59), полученный ранее в контексте линейной регрессии. Напомним, что для локализованных базисных функций прогностическая дисперсия для моделей линейной регрессии становится малой в областях исходного пространства, где нет базисных функций. В случае RVM с базисными функциями, центрированными в точках исходных данных, модель будет становиться все более уверенной в своих прогнозах при экстраполяции вне области исходных данных (Rasmussen and Qui.nonero-Candela, 2005), что, разумеется, нежелательно (*см. раздел 6.4.2*). Прогностическое распределение в регрессии на основе гауссовского процесса не имеет этого недостатка. Однако вычислительная стоимость составления прогнозов с гауссовскими процессами обычно намного выше, чем для метода RVM.

На рис. 7.9 приведен пример применения метода RVM к набору данных для синусоидальной регрессии. Здесь параметр точности шума β также определяется путем максимизации правдоподобия. Мы видим, что количество релевантных векторов в методе RVM значительно меньше количества опорных векторов, используемых в методе SVM. Обнаружено, что для широкого диапазона задач регрессии и классификации метод RVM дает модели, которые, как правило, на порядок более компактны, чем соответствующая модель, построенная с помощью метода опорных векторов, что приводит к значительному улучшению скорости обработки тестовых данных. Примечательно, что эта большая разреженность достигается с небольшой ошибкой обобщения (или вообще ее отсутствием) по сравнению с соответствующим SVM.

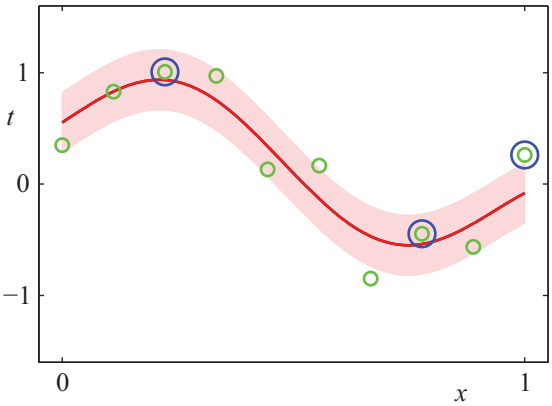


Рис. 7.9. Регрессия с помощью метода RVM на основе того же набора данных и тех же гауссовских ядер, что и на рис. 7.8 для модели регрессии ν-SVM. Математическое ожидание прогностического распределения в методе RVM показано красной кривой, а одно стандартное отклонение прогностического распределения — заштрихованной областью. Кроме того, точки данных показаны зеленым цветом, а релевантные векторы — синими кружками. Обратите внимание на то, что в модели есть только три релевантных вектора по сравнению с семью опорными векторами в методе ν-SVM на рис. 7.8

Основным недостатком метода RVM по сравнению с методом SVM является то, что обучение предполагает оптимизацию невыпуклой функции, а время обучения может быть больше, чем для сопоставимого варианта SVM. Для модели с M базисными функциями для RVM требуется обращение матрицы размера $M \times M$, которая, как правило, требует выполнения порядка $O(M^3)$ вычислительных операций. В конкретном случае SVM-подобной модели (7.78) имеем: $M = N + 1$. Как мы уже отмечали, существуют методы обучения SVM, стоимость которых примерно квадратична по N. Конечно, в случае RVM мы всегда имеем возможность начать с меньшего количества базисных функций, чем $N + 1$. Более важно то, что в методе релевантных векторов параметры, определяющие сложность, и дисперсия шума определяются автоматически за один сеанс обучения, тогда как в методе опорных векторов параметры C и ε (или ν) обычно обнаруживаются с использованием перекрестной проверки, которая включает в себя несколько сеансов обучения. Кроме того, в следующем разделе мы выведем альтернативную процедуру обучения метода опорных векторов, которая значительно улучшит скорость обучения.

7.2.2. Анализ разреженности

Ранее мы отметили, что механизм автоматического определения релевантности приводит к тому, что некоторое подмножество параметров приводится к нулю. Теперь мы более подробно рассмотрим механизм разреженности в контексте метода релевантных векторов. По ходу дела мы придем к значительно более быстрой процедуре оптимизации гиперпараметров по сравнению с приведенными выше прямыми методами.

Прежде чем приступить к анализу, сформулируем неофициальное представление о происхождении разреженности в байесовских линейных моделях. Рассмотрим множество данных, содержащий $N = 2$ наблюдений t_1 и t_2, вместе с моделью, имеющей единственную базисную функцию $\phi(\mathbf{x})$, с гиперпараметром α наряду с изотропными шумами, имеющими точность β. Из (7.85) следует, что маргинальное правдоподобие задается выражением $p(\mathbf{t}|\alpha, \beta) = \mathcal{N}(\mathbf{t}|\mathbf{0}, \mathbf{C})$, в котором ковариационная матрица принимает вид

$$C = \frac{1}{\beta}\mathbf{I} + \frac{1}{\alpha}\boldsymbol{\varphi}\boldsymbol{\varphi}^{\mathrm{T}}, \tag{7.92}$$

где $\boldsymbol{\varphi}$ обозначает N-мерный вектор $(\phi(\mathbf{x}_1), \phi(\mathbf{x}_2))^{\mathrm{T}}$ и аналогично $\mathbf{t} = (t_1, t_2)^{\mathrm{T}}$. Обратите внимание на то, что это всего лишь модель гауссовского процесса с нулевым математическим ожиданием по $\mathbf{t}$ с ковариационной матрицей $\mathbf{C}$. Наша цель — при заданном наблюдении $\mathbf{t}$ найти α^* и β^*, максимизируя маргинальное правдоподобие. На рис. 7.10 видно, что, если между направлением $\boldsymbol{\varphi}$ и вектором $\mathbf{t}$ обучающих данных существует плохая согласованность, соответствующий гиперпараметр α будет стремиться к ∞, а базисный вектор будет исключен из модели. Этот эффект возникает из-за того, что любое конечное значение для α всегда будет определять меньшую вероятность данных, тем самым уменьшая значение плотности при $\mathbf{t}$, при условии, что гиперпараметр β имеет оптимальное значение. Мы видим, что любое конечное значение для α приведет к тому, что распределение будет продолжено в направлении, удаляющемся от данных, тем самым увеличивая массу вероятности в областях, расположенных далеко от наблюдаемых данных, и, следовательно, уменьшая значение плотности на самом целевом объекте данных. Для более общего случая M базисных векторов $\boldsymbol{\varphi}_1, ..., \boldsymbol{\varphi}_M$ имеет место аналогичная ситуация, а именно: если конкретный базисный вектор плохо согласован с вектором данных $\mathbf{t}$, то он, вероятно, будет исключен из модели.

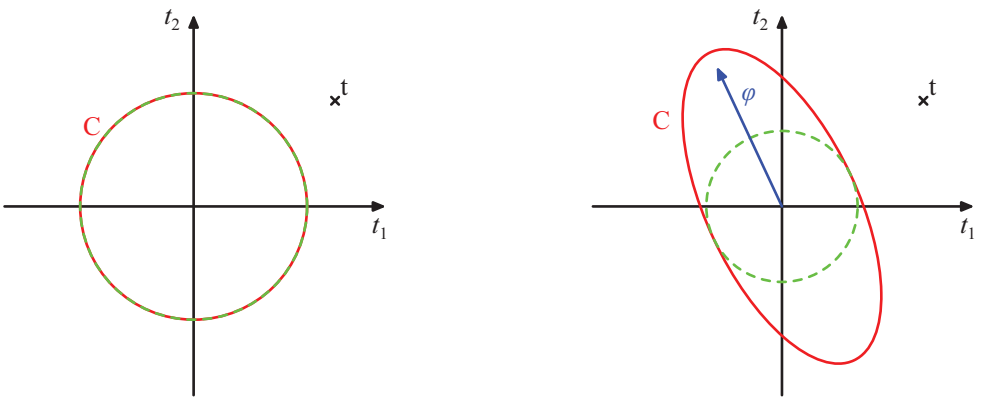


Рис. 7.10. Механизм разреженности в байесовской линейной регрессионной модели с обучающим вектором целевых значений $\mathbf{t} = (t_1, t_2)^{\mathrm{T}}$, обозначенным крестиком, для модели с одним базисным вектором $\boldsymbol{\varphi} = (\phi(\mathbf{x}_1), \phi(\mathbf{x}_2))^{\mathrm{T}}$, которая слабо согласована с целевым вектором $\mathbf{t}$. Слева показана модель, имеющая только изотропный шум, так что $\mathbf{C} = \beta^{-1}\mathbf{I}$, что соответствует $\alpha = \infty$, причем гиперпараметр β принимает свое наиболее вероятное значение. Справа показана та же модель, но с конечным значением α. В каждом случае красный эллипс соответствует единичному расстоянию Махаланобиса, причем на обоих рисунках $|\mathbf{C}|$ принимает одинаковое значение, в то время как пунктирный зеленый круг демонстрирует вклад, возникающий из-за шума β^{-1}. Мы видим, что любое конечное значение α уменьшает вероятность наблюдаемых данных, поэтому для нахождения наиболее вероятного решения базисный вектор исключается из модели

Теперь исследуем механизм разреженности с более строгой математической точки зрения для общего случая с M базисных функций. Чтобы обосновать этот анализ, прежде всего отметим, что в формуле (7.87) для уточнения параметра α_i члены в правой части сами по себе также являются функциями, зависящими от α_i. Таким образом, эта формула являются неявной, и итерация потребуется даже для определения единственного α_i при фиксированных значениях всех остальных α_j при $j \neq i$.

Следовательно, необходимо искать другой подход к оптимизации RVM, в которой явно определяется зависимость маргинального правдоподобия (7.85) от конкретного α_i, а затем явно определяются его стационарные точки (Faul and Tipping, 2002; Faul, 2003). Для этого сначала выведем вклад α_i в матрицу C, определяемую формулой (7.86):

$$\begin{aligned}\mathbf{C} &= \beta^{-1}\mathbf{I} + \sum_{j\neq i}\alpha_j^{-1}\boldsymbol{\varphi}_j\boldsymbol{\varphi}_j^{\mathrm{T}} + \alpha_i^{-1}\boldsymbol{\varphi}_i\boldsymbol{\varphi}_i^{\mathrm{T}} = \\ &= \mathbf{C}_{-i} + \alpha_j^{-1}\boldsymbol{\varphi}_i\boldsymbol{\varphi}_i^{\mathrm{T}},\end{aligned} \tag{7.93}$$

где $\boldsymbol{\varphi}_i$ обозначает i-й столбец матрицы $\boldsymbol{\Phi}$, иначе говоря, N-мерный вектор с элементами $(\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_N))$, в отличие от вектора $\boldsymbol{\phi}_n$, который обозначает n-ю строку матрицы $\boldsymbol{\Phi}$. Матрица $\mathbf{C}_{-i}$ представляет собой матрицу $\mathbf{C}$ с удаленной i-й базисной функцией. Используя матричные тождества (B.7) и (B.15), определитель и матрицу, обратную матрице $\mathbf{C}$, можно записать:

$$|\mathbf{C}| = |\mathbf{C}_{-1}|\left(1 + \alpha_i^{-1}\varphi_i^{\mathrm{T}}\mathbf{C}_{-i}^{-1}\varphi_i\right), \tag{7.94}$$

$$\mathbf{C}^{-1} = \mathbf{C}_{-i}^{-1} - \frac{\mathbf{C}_{-i}^{-1}\boldsymbol{\varphi}_i\boldsymbol{\varphi}_i^{\mathrm{T}}\mathbf{C}_{-i}^{-1}}{\alpha_i + \boldsymbol{\varphi}_i^{\mathrm{T}}\mathbf{C}_{-i}^{-1}\boldsymbol{\varphi}_i}. \tag{7.95}$$

Используя эти результаты, можно написать функцию маргинального правдоподобия (7.85) в виде (***см. упражнение 7.15***)

$$L(\boldsymbol{\alpha}) = L(\boldsymbol{\alpha}_{-i}) + \lambda(\alpha_i), \tag{7.96}$$

где $L(\boldsymbol{\alpha}_{-i})$ — логарифмическое маргинальное правдоподобие с исключенной базисной функцией $\boldsymbol{\varphi}_i$, а величина $\lambda(\alpha_i)$ определяется формулой

$$\lambda(\alpha_i) = \frac{1}{2}\left[\ln\alpha_i - \ln(\alpha_i + s_i) + \frac{q_i^2}{\alpha_i + s_i}\right] \tag{7.97}$$

и выражает всю зависимость от α_i. Здесь мы ввели две величины:

$$s_i = \boldsymbol{\varphi}_i^{\mathrm{T}}\mathbf{C}_{-i}^{-1}\boldsymbol{\varphi}_i, \tag{7.98}$$

$$q_i = \boldsymbol{\varphi}_i^{\mathrm{T}}\mathbf{C}_{-i}^{-1}\mathbf{t}. \tag{7.99}$$

где s_i называется *разреженностью*, а q_i — *качеством* базисной функции $\boldsymbol{\varphi}_i$, и, как мы увидим, большое значение s_i по отношению к значению q_i означает, что базисная функция $\boldsymbol{\varphi}_i$ скорее всего будет исключена из модели. Разреженность измеряет степень перекрытия базисной функции $\boldsymbol{\varphi}_i$ с другими базисными векторами в модели, а качество представляет собой меру согласованности базисного вектора $\boldsymbol{\varphi}_i$ с разностью между обучающими целевыми значениями $\mathbf{t} = (t_1, \ldots, t_N)^{\mathrm{T}}$ и вектором прогнозов $\mathbf{y}_{-i}$, которые были бы результатом модели с исключенным вектором $\boldsymbol{\varphi}_i$ (Tipping and Faul, 2003).

Стационарные точки маргинального правдоподобия относительно α_i возникают, когда производная

$$\frac{d\lambda(\alpha_i)}{d\alpha_i} = \frac{\alpha_i^{-1}s_i^2 - \left(q_i^2 - s_i\right)}{2\left(\alpha_i + s_i\right)^2} \tag{7.100}$$

равна нулю. Решение имеет две возможные формы. Вспоминая, что $\alpha_i \geq 0$, мы видим, что если $q_i^2 < s$, то получаем решение при $\alpha_i \to \infty$. И наоборот, если $q_i^2 > s$, можно решить уравнение относительно α_i:

$$\alpha_i = \frac{s_i^2}{q_i^2 - s_i}. \tag{7.101}$$

Эти два решения показаны на рис. 7.11. Мы видим, что относительная величина качества и разреженности определяет, будет ли конкретный базисный вектор исключаться из модели или нет. Более полный анализ (Faul and Tipping, 2002), основанный на вторых производных маргинального правдоподобия, подтверждает, что эти решения действительно являются единственными максимумами $\lambda(\alpha_i)$ (***см. упражнение 7.16***).

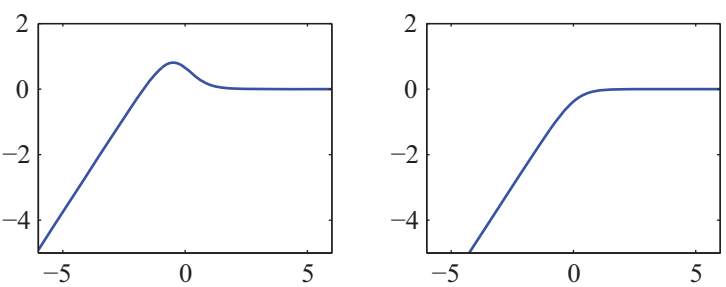


Рис. 7.11. Графики логарифмического маргинального правдоподобия $\lambda(\alpha_i)$ от $\ln \alpha_i$, имеющие единственный максимум при конечном α_i для $q_i^2 = 4$ и $s_i = 1$ (так, что $q_i^2 > s_i$), слева на рисунке, и максимум при $\alpha_i \to \infty$ для $q_i^2 = 1$ и $s_i = 2$ (так, что $q_i^2 < s_i$) — справа

Заметим, что этот подход позволил получить замкнутое решение для α_i при заданных значениях других гиперпараметров. Этот анализ также позволяет получить представление о происхождении разреженности в методе RVM, что также приводит к практическому алгоритму оптимизации гиперпараметров, обладающих значительными скоростными преимуществами. Он использует фиксированный набор потенциальных векторов-кандидатов, а затем циклически проходит через них, чтобы решить, должен ли каждый вектор быть включен в модель. Полученный последовательный разреженный байесовский алгоритм обучения описан ниже.

Последовательный разреженный байесовский алгоритм обучения

1. Если решается задача регрессии, то задайте начальное значение β.
2. Инициализируйте базисную функцию $\boldsymbol{\varphi}_1$, задав гиперпараметр α_1 по формуле (7.101), а остальные гиперпараметры α_j при $j \neq 1$ — бесконечным значением, так чтобы в модель входила только функция $\boldsymbol{\varphi}_1$.
3. Вычислите матрицу $\boldsymbol{\Sigma}$ и вектор $\mathbf{m}$, а также q_i и s_i для всех базисных функций.
4. Выберите базисную функцию-кандидат $\boldsymbol{\varphi}_i$.
5. Если $q_i^2 > s_i$ и $\alpha_i < \infty$, так что базисный вектор $\boldsymbol{\varphi}_i$ уже включен в модель, то обновите α_i, используя (7.101).
6. Если $q_i^2 > s_i$ и $\alpha_i = \infty$, включите $\boldsymbol{\varphi}_i$ в модель и оцените гиперпараметр α_i, используя (7.101).
7. Если $q_i^2 \leq s_i$ и $\alpha_i < \infty$, то удалите базисную функцию $\boldsymbol{\varphi}_i$ из модели и задайте $\alpha_i = \infty$.
8. При решении задачи регрессии обновите β.
9. Если процесс сошелся, завершите выполнение, в противном случае перейдите к п. 3.

Заметим, что если $q_i^2 \leq s_i$ и $\alpha_i = \infty$, то базисная функция $\boldsymbol{\varphi}_i$ уже исключена из модели и никаких действий не требуется. На практике удобно вычислять величины

$$Q_i = \boldsymbol{\varphi}_i^{\mathrm{T}} \mathbf{C}^{-1} \mathbf{t}, \tag{7.102}$$

$$S_i = \boldsymbol{\varphi}_i^{\mathrm{T}} \mathbf{C}^{-1} \boldsymbol{\varphi}_i, \tag{7.103}$$

В таком случае переменные качества и разреженности могут быть выражены в форме

$$q_i = \frac{\alpha_i Q_i}{\alpha_i - S_i}, \tag{7.104}$$

$$s_i = \frac{\alpha_i S_i}{\alpha_i - S_i}. \tag{7.105}$$

Заметим, что при $\alpha_i = \infty$ имеем: $q_i = Q_i$ и $s_i = S_i$ (***см. упражнение 7.17***). Используя (B.7), можно записать:

$$Q_i = \beta \boldsymbol{\varphi}_i^{\mathrm{T}} \mathbf{t} - \beta^2 \boldsymbol{\varphi}_i^{\mathrm{T}} \boldsymbol{\Phi} \boldsymbol{\Sigma} \boldsymbol{\Phi}^{\mathrm{T}} \mathbf{t}, \tag{7.106}$$

$$S_i = \beta \boldsymbol{\varphi}_i^{\mathrm{T}} \boldsymbol{\varphi}_i - \beta^2 \varphi_i^{\mathrm{T}} \boldsymbol{\Phi} \boldsymbol{\Sigma} \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\varphi}_i, \tag{7.107}$$

где матрицы $\boldsymbol{\Phi}$ и $\boldsymbol{\Sigma}$ содержат только те базисные векторы, которые соответствуют конечным гиперпараметрам α_i. Следовательно, на каждом этапе требуемые вычисления имеют порядок $O(M^3)$, где M — количество активных базисных векторов в модели, которое, как правило, намного меньше, чем количество N обучающих образов.

7.2.3. Метод RVM для классификации

Мы можем распространить метод релевантных векторов на задачи классификации, применяя априорное распределение ARD по весам к вероятностной модели линейной классификации, изученной в главе 4. Сначала рассмотрим бинарную задачу классификации с целевой переменной $t \in \{0, 1\}$. Теперь модель принимает форму линейной комбинации базисных функций, преобразуемых логистической сигмоидой:

$$y(\mathbf{x}, \mathbf{w}) = \sigma(\mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x})), \tag{7.108}$$

где $\sigma(\cdot)$ — логистическая сигмоида, определяемая формулой (4.59). Если мы введем нормальное априорное распределение по весовому вектору $\mathbf{w}$, то получим модель, которая рассмотрена в главе 4. Разница в том, что в методе RVM эта модель использует априорное распределение ARD (7.80), в которой есть отдельный гиперпараметр точности, связанный с каждым весовым параметром.

В отличие от модели регрессии, мы больше не можем аналитически интегрировать вектор параметров $\mathbf{w}$. Здесь мы следуем Tipping (2001) и используем аппроксимацию Лапласа (***см. раздел 4.4***), которая была применена к тесно связанной задаче байесовской логистической регрессии в ***разделе 4.5.1***.

Начнем с инициализации вектора гиперпараметров $\boldsymbol{\alpha}$. Затем при заданном значении $\boldsymbol{\alpha}$ построим гауссовскую аппроксимацию апостериорного распределения и тем самым получим приближение к маргинальному правдоподобию. После этого максимизация приближенного маргинального правдоподобия приводит к уточнению значения $\boldsymbol{\alpha}$, и процесс повторяется, пока не сойдется.

Рассмотрим подробнее аппроксимацию Лапласа для этой модели. При фиксированном значении $\boldsymbol{\alpha}$ мода апостериорного распределения по $\mathbf{w}$ вычисляется путем максимизации функции

$$\begin{aligned}\ln p\left(\mathbf{w}|\mathbf{t},\boldsymbol{\alpha}\right) &= \ln\left\{p\left(\mathbf{t}|\mathbf{w}\right)p\left(\mathbf{w}|\boldsymbol{\alpha}\right)\right\} - \ln p\left(\mathbf{t}|\boldsymbol{\alpha}\right) = \\ &= \sum_{n=1}^{N}\left\{t_n \ln y_n + \left(1-t_n\right)\ln\left(1-y_n\right)\right\} - \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{A}\mathbf{w} + const,\end{aligned} \tag{7.109}$$

где $\mathbf{A} = \mathrm{diag}(\alpha_i)$. Это можно сделать, используя метод наименьших квадратов с итеративным пересчетом весов (IRLS), как описано в ***разделе 4.3.3***. Для этого нам нужен вектор градиента и матрица Гессе (***см. упражнение 7.18***) для логарифма апостериорного распределения (7.109):

$$\nabla \ln p(\mathbf{w}|\mathbf{t},\ \boldsymbol{\alpha}) = \boldsymbol{\Phi}^{\mathrm{T}}(\mathbf{t} - \mathbf{y}) - \mathbf{A}\mathbf{w}, \tag{7.110}$$

$$\nabla\nabla \ln p(\mathbf{w}|\mathbf{t},\ \boldsymbol{\alpha}) = -\left(\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{B}\boldsymbol{\Phi} + \mathbf{A}\right), \tag{7.111}$$

где $\mathbf{B}$ — диагональная матрица $N \times N$ с элементами $b_n = y_n(1-y_n)$; вектор $\mathbf{y} = (y_1, ..., y_N)^{\mathrm{T}}$; $\boldsymbol{\Phi}$ — матрица плана с элементами $\Phi_{ni} = \phi_i(\mathbf{x}_n)$. Здесь мы использовали свойство производной логистической сигмоиды (4.88). При условии сходимости алгоритма IRLS отрицательный гессиан представляет собой обратную ковариационную матрицу для гауссовской аппроксимации апостериорного распределения.

Мода результирующей аппроксимации апостериорного распределения, соответствующая математическому ожиданию гауссовской аппроксимации, получается путем приравнивания (7.110) к нулю, что дает математическое ожидание и ковариантную матрицу аппроксимации Лапласа в виде

$$\mathbf{w}^* = \mathbf{A}^{-1}\boldsymbol{\Phi}^{\mathrm{T}}(\mathbf{t} - \mathbf{y}), \tag{7.112}$$

$$\boldsymbol{\Sigma} = \left(\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{B}\boldsymbol{\Phi} + \mathbf{A}\right)^{-1}. \tag{7.113}$$

Теперь мы можем использовать эту аппроксимацию Лапласа для вычисления маргинального правдоподобия. Используя общий результат (4.135) для интеграла, вычисленного с использованием аппроксимации Лапласа, имеем:

$$\begin{aligned}p\left(\mathbf{t}|\boldsymbol{\alpha}\right) &= \int p\left(\mathbf{t}|\mathbf{w}\right)p\left(\mathbf{w}|\boldsymbol{\alpha}\right)d\mathbf{w} \simeq \\ &\simeq p\left(\mathbf{t}|\mathbf{w}^*\right)p\left(\mathbf{w}^*\middle|\boldsymbol{\alpha}\right)\left(2\pi\right)^{M/2}\left|\boldsymbol{\Sigma}\right|^{1/2}.\end{aligned} \tag{7.114}$$

Если подставить $p(\mathbf{t}|\mathbf{w}^*)$ и $p(\mathbf{w}^*|\boldsymbol{\alpha})$, а затем приравнять к нулю производную от маргинального правдоподобия по α, то получим (***см. упражнение 7.19***):

$$-\frac{1}{2}\left(w_i^*\right)^2+\frac{1}{2\alpha_i}-\frac{1}{2}\Sigma_{ii}=0. \tag{7.115}$$

Определив $\gamma_i = 1 - \alpha_i\Sigma_{ii}$ и выполняя перестановку, получим формулу

$$\alpha_i^{new}=\frac{\gamma_i}{\left(w_i^*\right)^2}, \tag{7.116}$$

идентичную формуле уточнения (7.87), полученной для регрессии RVM.

Если мы определим

$$\hat{\mathbf{t}}=\boldsymbol{\Phi}\mathbf{w}^*+\mathbf{B}^{-1}\left(\mathbf{t}-\mathbf{y}\right), \tag{7.117}$$

то сможем записать приближенную логарифмическую функцию маргинального правдоподобия в виде

$$\ln p\left(\mathbf{t}|\boldsymbol{\alpha}\right)=-\frac{1}{2}\left\{N\ln\left(2\pi\right)+\ln\left|\mathbf{C}\right|+\left(\hat{\mathbf{t}}\right)^{\mathrm{T}}\mathbf{C}^{-1}\hat{\mathbf{t}}\right\}, \tag{7.118}$$

где

$$\mathbf{C}=\mathbf{B}+\boldsymbol{\Phi}\mathbf{A}\boldsymbol{\Phi}^{\mathrm{T}}. \tag{7.119}$$

Эта матрица имеет ту же форму, что и (7.85) в случае регрессии, поэтому можем применить тот же анализ разреженности и получить тот же алгоритм быстрого обучения, в котором мы полностью оптимизируем один гиперпараметр α_i на каждом шаге.

На рис. 7.12 продемонстрировано применение метода релевантных векторов к искусственному множеству данных. Мы видим, что релевантные векторы, как правило, не лежат в области границы решения, в отличие от метода опорных векторов. Это согласуется с предыдущим обсуждением разреженности в методе RVM, поскольку базисная функция $\phi_i(\mathbf{x})$, центрированная в точке данных вблизи границы, будет соответствовать вектору $\boldsymbol{\varphi}_i$, который плохо согласован с вектором обучающих данных $\mathbf{t}$.

Одним из потенциальных преимуществ метода релевантных векторов по сравнению с SVM является то, что он дает вероятностные прогнозы. Например, это позволяет использовать RVM, чтобы помочь построить плотность излучения в нелинейном расширении линейной динамической системы (***см. раздел 13.3***) для отслеживания граней в видеопоследовательностях (Williams *et al.*, 2005). До сих пор мы рассматривали RVM для задач бинарной классификации. Для $K > 2$ классов мы снова используем вероятностный подход из ***раздела 4.3.4***, в котором существуют K линейных моделей вида

$$a_k=\mathbf{w}_k^{\mathrm{T}}\mathbf{x}, \tag{7.120}$$

которые в сочетании с функцией softmax дают такие результаты:

$$y_k(\mathbf{x}) = \frac{\exp(a_k)}{\sum_j \exp(a_j)}. \tag{7.121}$$

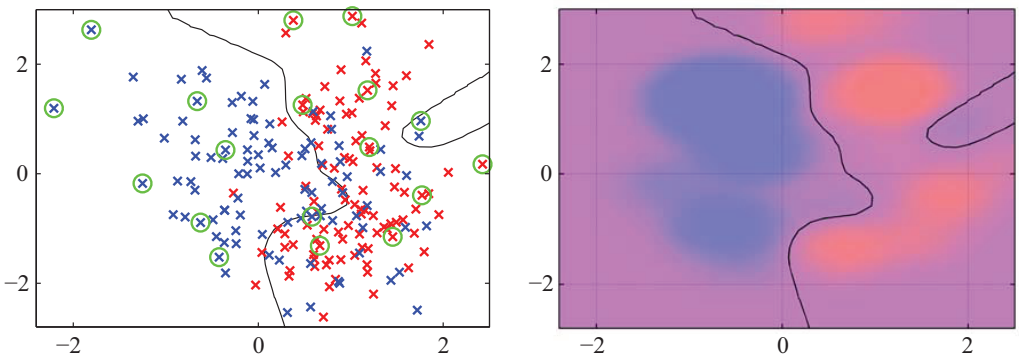


Рис. 7.12. Пример применения метода релевантных векторов к искусственному набору данных, в котором на левом графике показана граница решения и точки данных вместе с релевантными векторами, обозначенными кружочками. Сравнение с результатами, приведенными на рис. 7.4 для соответствующего метода опорных векторов, показывает, что RVM дает намного более разреженную модель. Правый график показывает апостериорную вероятность, заданную выходом RVM, в которой доля красного (синего) цвета указывает вероятность того, что эта точка принадлежит красному (синему) классу

В таком случае логарифмическая функция правдоподобия задается формулой

$$\ln p\left(\mathbf{T}\middle|\mathbf{w}_1,\ldots,\mathbf{w}_K\right) = \prod_{n=1}^{N}\prod_{k=1}^{K} y_{nk}^{t_{nk}}, \tag{7.122}$$

где целевые значения t_{nk} каждой точки исходных данных n закодированы по схеме 1 из K, а $\mathbf{T}$ — матрица с элементами t_{nk}. Как и в предыдущих случаях, для оптимизации гиперпараметров можно использовать аппроксимацию Лапласа (Tipping, 2001), в которой модель и ее матрица Гессе вычисляются с помощью метода IRLS. Это создает более принципиальный подход к классификации многих классов по сравнению с попарным методом, используемым в методе опорных векторов, а также дает вероятностные прогнозы для новых точек. Основным недостатком является то, что матрица Гессе имеет размер $MK \times MK$, где M — количество активных базисных функций, что дает дополнительный коэффициент K^3 в вычислительной стоимости обучения по сравнению с двухклассовым методом RVM.

Основным недостатком метода релевантных векторов является относительно долгое время обучения по сравнению с SVM. Однако это компенсируется отсутствием сеансов перекрестной проверки для определения параметров сложности модели. Кроме того, поскольку этот метод дает более разреженные модели, время вычисления на тестовых точках, которое обычно более важно на практике, как правило, намного меньше. чем у метода SVM.

Упражнения

7.1. (**) WWW Предположим, у нас есть множество входных векторов $\{\mathbf{x}_n\}$ с соответствующими целевыми значениями $t_n \in \{-1, 1\}$ и мы моделируем плотность входных векторов в каждом классе отдельно, используя плотность ядра Парзена (*см. раздел 2.5.1*) с ядром $k(\mathbf{x}, \mathbf{x}')$. Сформулируйте правило принятия решения с минимальным уровнем ошибок, предполагая, что два класса имеют одинаковую вероятность. Покажите, что если ядро выбрано в виде $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^{\mathrm{T}}\mathbf{x}'$, то правило классификации сводится к простому присвоению нового входного вектора классу, имеющему самое близкое математическое ожидание. Наконец, покажите, что если ядро имеет вид $k(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}')$, то классификация основана на самом близком математическом ожидании в пространстве признаков $\boldsymbol{\phi}(\mathbf{x})$.

7.2. (*) Покажите, что если единицу в правой части ограничения (7.5) заменить некоторой произвольной постоянной $\gamma > 0$, то гиперплоскость с максимальным зазором не изменится.

7.3. (**) Покажите, что независимо от размерности исходного пространства для определения местоположения гиперплоскости с максимальным зазором достаточно множества, состоящего всего из двух точек, по одной от каждого класса.

7.4. (**) WWW Покажите, что ширина максимального зазора ρ определяется выражением

$$\frac{1}{\rho^2} = \sum_{n=1}^{N} a_n, \tag{7.123}$$

где числа $\{a_n\}$ определяются с помощью максимизации (7.10) с учетом ограничений (7.11) и (7.12).

7.5. (**) Покажите, что значения ρ и $\{a_n\}$ в предыдущем упражнении также удовлетворяют условию

$$\frac{1}{\rho^2} = 2\tilde{L}(\mathbf{a}), \tag{7.124}$$

где функция $\tilde{L}(\mathbf{a})$ определяется формулой (7.10). Аналогичным образом покажите, что

$$\frac{1}{\rho^2} = \|\mathbf{w}\|^2. \tag{7.125}$$

7.6. (*) Рассмотрим модель логистической регрессии с целевой переменной $t \in \{-1, 1\}$. Покажите, что, если мы определим $p(t = 1|y) = \sigma(y)$, где $y(\mathbf{x})$ задается формулой (7.1), отрицательный логарифм правдоподобия с добавлением квадратичного члена регуляризации примет вид (7.47).

7.7. (*) Рассмотрим лагранжиан (7.56) для регрессионного метода опорных векторов. Приравняв к нулю производные лагранжиана по $\mathbf{w}$, b, ξ_n и $\hat{\xi}_n$, а затем выполнив обратную подстановку для исключения соответствующих переменных, покажите, что двойственный лагранжиан задается формулой (7.61).

7.8. (*) **WWW** Для регрессионного метода опорных векторов, рассмотренного в ***разделе 7.1.4***, покажите, что все точки обучающих данных, для которых $\xi_n > 0$, удовлетворяют условию $a_n = C$, и аналогично, все точки, для которых $\hat{\xi}_n > 0$, удовлетворяют условию $\hat{a}_n > C$.

7.9. (*) Проверьте результаты (7.82) и (7.83) для математического ожидания и ковариационной матрицы апостериорного распределения по весам в регрессионном методе RVM.

7.10. (**) **WWW** Выведите результат (7.85) для маргинального правдоподобия в регрессионном методе RVM, выполнив интегрирование нормального распределения по $\mathbf{w}$ в (7.84) с помощью выделения полного квадрата в экспоненте.

7.11. (**) Повторите вышеуказанное упражнение, но на этот раз используя общий результат (2.115).

7.12. (**) **WWW** Покажите, что прямая максимизация логарифмической функции маргинального правдоподобия (7.85) для регрессионного метода релевантных векторов приводит к уравнениям уточнения гиперпараметров (7.87) и (7.88), где γ_i определяется по формуле (7.89).

7.13. (**) Анализируя регрессионный RVM, мы получили формулы уточнения гиперпараметров (7.87) и (7.88) путем максимизации маргинального правдоподобия, заданного формулой (7.85). Расширьте этот подход, включив априорные распределения гиперпараметров, заданные гамма-распре-

делениями вида (Б.26), и получите соответствующие формулы уточнения для $\boldsymbol{\alpha}$ и β, максимизируя соответствующую апостериорную вероятность $p(\mathbf{t}, \boldsymbol{\alpha}, \beta \mid \mathbf{x})$ по $\boldsymbol{\alpha}$ и β.

7.14. (**) Выведите результат (7.90) для прогностического распределения в регрессионном методе релевантных векторов. Покажите, что прогностическая дисперсия задается формулой (7.91).

7.15. (**) **WWW** Используя результаты (7.94) и (7.95), покажите, что маргинальное правдоподобие (7.85) можно записать в виде (7.96), где $\lambda(\alpha_n)$ определяется формулой (7.97), а коэффициенты разреженности и качества определяются формулами (7.98) и (7.99) соответственно.

7.16. (*) Вычислив вторую производную логарифмической функции маргинального правдоподобия (7.97) для регрессионного метода RVM относительно гиперпараметра α_i, покажите, что стационарная точка, заданная формулой (7.101), является максимумом маргинального правдоподобия.

7.17. (**) Используя (7.83) и (7.86) вместе с матричным тождеством (В.7), покажите, что величины S_n и Q_n, определенные формулами (7.102) и (7.103), можно записать в виде (7.106) и (7.107).

7.18. (*) **WWW** Покажите, что вектор градиента и матрица Гессе логарифма апостериорного распределения (7.109) в методе релевантных векторов для классификации задаются формулами (7.110) и (7.111).

7.19. (**) Убедитесь, что максимизация аппроксимации маргинального правдоподобия (7.114) в методе релевантных векторов для классификации приводит к результату (7.116) для уточнения гиперпараметров.