Лекція 7 [1]

Метод опорних векторів

У результаті інтенсивних наукових досліджень в області машинного навчання, спрямованих на підвищення якості класифікаторів, з'явилося нове покоління методів, зокрема — методи опорних векторів (Support Vector Machines — SVM), бустинга дерев рішень (boosted decision trees), регуляризованої логістичної регресії (regularized logistic regression), нейронних мереж (neural networks) і випадкових лісів (random forests). Багато з цих методів, включаючи метод опорних векторів, описаний у цій лекції, з успіхом застосовувалися для розв'язання задач інформаційного пошуку, зокрема при класифікації текстів. Метод опорних векторів являє собою різновид класифікатора "із широким зазором": він належить до методів машинного навчання, заснованих на моделі векторного простору, мета яких — знайти поділяючі поверхні між класами, максимально віддалені від усіх точок навчальної множини (можливо, проігнорувавши деякі точки як викиди чи шум).

Спочатку ми опишемо варіант методу опорних векторів для випадку двох класів, що допускають поділ за допомогою лінійного класифікатора (розділ 7.1), а потім розширимо цю модель на нероздільні дані, задачі з декількома класами і нелінійні задачі, а також наведемо деякі факти, що стосуються ефективності цього методу (розділ 7.2).

7.1. Метод опорних векторів: випадок лінійно роздільних класів

Якщо навчальна множина містить два класи даних, що допускають лінійний поділ, то існує велика кількість лінійних класифікаторів, за допомогою яких можна розділити ці дані. Інтуїтивно ясно, що поділяюча поверхня, яка проходить через середину смуги, що розділяє два класи, краще, ніж поділяюча поверхня, що лежить дуже близько до екземплярів одного чи обох класів. У той час як одні методи навчання, такі як персептрон, дозволяють знайти хоча б один лінійний роздільник, інші методи, такі як наївний байєсівський метод, знаходять найкращий лінійний роздільник, використовуючи визначений критерій. Зокрема, метод опорних векторів шукає поділяючу поверхню, максимально віддалену від будь-яких точок даних. Відстань між цією поверхнею і найближчою точкою даних називається *зазором класифікатора*. У методі опорних векторів обов'язково мається на увазі, що вирішальна функція цілком визначається (звичайно малою) підмножиною даних, що впливають на положення роздільника. Ці точки називаються *опорними векторами*, тому що у векторному просторі точку можна розглядати як вектор між початком координат і цією точкою. Зазор і опорні вектори для простої задачі показані на рис. 7.1. Інші точки даних не впливають на вибір поділяючої поверхні.

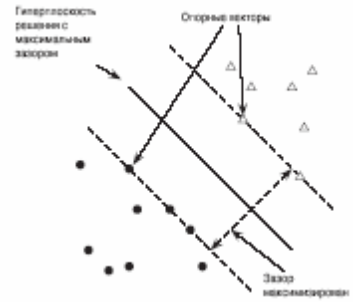

Рис. 7.1. Опорні вектори: п'ять точок, що лежать на межі зазору

Максимізація зазору виглядає гарною ідеєю, оскільки точки, що лежать поблизу поділяючою поверхні, породжують велику невизначеність; з імовірністю 50% класифікатор може прийняти кожне з двох рішень. Класифікатор з великим зазором знижує невизначеність рішення. Тим самим він створює визначений запас надійності: невелика помилка виміру чи невелика зміна документа не призведе до неправильної класифікації. Інше інтуїтивне обґрунтування методу опорних векторів продемонстроване на рис. 7.2. По своїй конструкції класифікатор SVM вимагає, щоб навколо поділяючої поверхні був широкий зазор. Якщо спробувати помістити між класами широку смугу, то діапазон кутів, при якому це можна зробити, виявиться набагато меншим, чим для гіперплощини. У результаті ємність запам'ятовування моделі зменшується, і можна чекати, що здатність моделі правильно узагальнювати тестові дані збільшується (див. обговорення компромісу між зсувом і дисперсією в лекції 7).

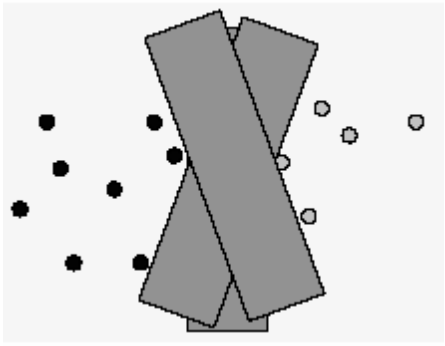

Рис. 7.2. Інтуїтивно зрозуміле обґрунтування класифікації із широким зазором. Прагнення до широкого зазору зменшує обсяг запам'ятовування моделі: діапазон кутів, при якому між двома множинами можна помістити широку поділяючу смугу, менше, ніж для поділяючої гіперплощини

Приведемо формальний алгебраїчний опис методу опорних векторів. Поділяюча гіперплощина задається параметром зсуву *b* (точкою перетинання з віссю *x)* і вектором $\vec{w}$ нормалі до поділяючого гіперплощині $\vec{w}$. У літературі по методах машинного навчання цей вектор звичайно називається *вектором весов* (weight vector). Для того щоб серед усіх гіперплощин, перпендикулярних вектору нормалі, вибрати одну потрібну гіперплощину, використовується параметр *b*. Оскільки поділяюча гіперплощина перпендикулярна вектору нормалі, усі точки $\vec{x}$ на гіперплощині задовольняють рівнянню $\vec{w}^T\vec{x} = -b$. Тепер допустимо, що в нас є навчальна множина $\mathbb{D} = \left\{\left(\vec{x}_i, y_i\right)\right\}$, у якому кожен елемент являє собою пару, що складається з точки $\vec{x}$ і відповідної мітки класу y_i. [1] У методі опорних векторів два класи завжди називаються +1 і -1 (а не 1 і 0), а *параметр зсуву* завжди явно позначається буквою *b* (а не включається у вектор $\vec{w}$ як константний доданок). Завдяки цьому математичні викладення стають набагато ясніше. У цьому випадку лінійний класифікатор описується наступною формулою.

$$f\left(\vec{x}\right) = sign\left(\vec{w}^T\vec{x} + b\right) \tag{7.1}$$

Значення –1 позначає один клас, а +1 — іншої.

Класифікація точки не викликає сумнівів, якщо вона лежить далеко від поділяючої поверхні. Для заданої сукупності даних і поділяючої гіперплощини *функціональним зазором* *i*-го екземпляру $\vec{x}_i$ стосовно гіперплощини $\left\langle \vec{w}, b \right\rangle$ називається величина $y_i\left(\vec{w}^T\vec{x}_i + b\right)$. У такому випадку функціональний зазор сукупності даних щодо поділяючої поверхні вдвічі більше функціонального зазору кожної з точок із сукупності даних з мінімальним функціональним зазором (множник 2 виникає за рахунок виміру всієї ширини зазору, як показано на рис. 7.3). Однак з використанням цього визначення зв'язана одна проблема: значення недостатнє обмежене, оскільки функціональний зазор можна зробити як завгодно великим, просто масштабуючи параметри $\vec{w}$ і *b*. Наприклад, якщо замінити вектор $\vec{w}$ вектором $5\vec{w}$, а параметр *b* — параметром 5*b*, то функціональний зазор збільшиться в п'ять разів: $y_i\left(5\vec{w}^T\vec{x}_i + 5b\right)$. Отже, необхідно якимсь образом обмежити величину вектора $\vec{w}$. Для цього необхідно згадати курс геометрії.

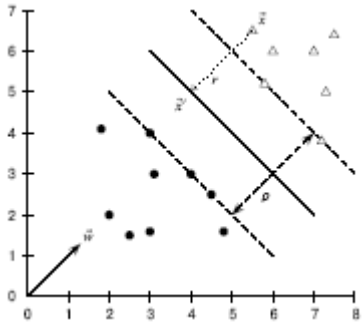

Рис. 7.3. Геометричний зазор точки r і поділяючої поверхні ρ

[1] Як вказувалося у лекції 7, ми розглядаємо точки у векторному просторі у самому загальному випадку, але якщо ці точки являють собою вектори документів, нормалізовані по довжині, то всі операції виконуються на поверхні сфери, і розділяюча поверхня її перетинає.

Що являє собою євклідова відстань між точкою $\vec{x}_i$ і поділяючою поверхнею? На рис. 7.3 воно позначено символом *r*. Як відомо, найкоротша відстань між точкою і гіперплощиною визначається перпендикуляром до площини, що, природно, паралельний вектору $\vec{w}$. Одиничний вектор у цьому напрямку має вид $\vec{w}/|\vec{w}|$. Пунктирна лінія на діаграмі являє собою паралельний перенос вектора $r\,\vec{w}/|\vec{w}|$. Позначимо точку, що лежить на гіперплощині, найближчу до вектора $\vec{x}$, через $\vec{x}'$. Таким чином,

$$\vec{x}' = \vec{x} - yr\frac{\vec{w}}{|\vec{w}|}\ . \tag{7.2}$$

Тут множення на число *y* просто змінює знак для двох положень вектора $\vec{x}$ по різні сторони поділяючої поверхні. Більш того, точка $\vec{x}'$ лежить на поверхні, а виходить, задовольняє рівнянню $\vec{w}^T\vec{x}' + b = 0$. Отже,

$$\vec{w}^T\left(\vec{x} - yr\frac{\vec{w}}{|\vec{w}|}\right) + b = 0\ . \tag{7.3}$$

Вирішуючи це рівняння відносно *r*, одержимо наступне рішення [2].

$$r = y\frac{\vec{w}^T\vec{x} + b}{|\vec{w}|} \tag{7.4}$$

Точки, найближчі до поділяючої гіперплощини, як і колись, є опорними векторами. *Геометричний зазор* — це максимальна ширина смуги, яку можна провести між опорними векторами двох класів. Інакше кажучи, це значення, що вдвічі перевищує мінімальне значення *r*, обчислене за формулою (7.4), чи, що еквівалентно, максимальна ширина однієї з роздільних смуг, показаних на рис. 7.2. Зовсім очевидно, що геометричний зазор не залежить від масштабування: заміна параметрів $\vec{w}$ на $5\vec{w}$ і *b* на 5*b* не призводить до зміни геометричного зазору, оскільки він нормалізується довжиною $\vec{w}$. Це значить, що ми можемо накласти на вектор $\vec{w}$ будь-які обмеження по масштабу, не впливаючи на геометричний зазор. Наприклад, можна установити обмеження $|\vec{w}| = 1$. У цьому випадку геометричний зазор збігається з функціональним.

Оскільки функціональний зазор можна довільно масштабувати, прагнучи до зручності розв'язування великих задач за допомогою методу опорних векторів, зажадаємо, щоб функціональний зазор усіх точок даних був не менше одиниці і дорівнював одиниці хоча б на одному векторі даних. Інакше кажучи, для всіх точок повинна виконуватися нерівність

$$y_i\left(\vec{w}^T\vec{x}_i + b\right) \geq 1\ , \tag{7.5}$$

і повинні існувати опорні вектори, на яких ця нерівність перетворюється в рівність. Оскільки відстань від точки $\vec{x}_i$ до гіперплощини дорівнює $r_i = y_i\left(\vec{w}^T\vec{x}_i + b\right)/|\vec{w}|$, геометричний зазор дорівнює $\rho = \dfrac{2}{|\vec{w}|}$. Наша мета — максимізувати геометричний зазор. Інакше кажучи, потрібно знайти параметри $\vec{w}$ і *b*, що задовольняють наступним умовам.

- Величина $\rho = \dfrac{2}{|\vec{w}|}$ досягає максимуму.
- При усіх $(\vec{x}_i, y_i) \in \mathbb{D}$ виконується умова $y_i\left(\vec{w}^T\vec{x}_i + b\right) \geq 1$.

Максимізація величини $\dfrac{2}{|\vec{w}|}$ еквівалентна мінімізації величини $\dfrac{|\vec{w}|}{2}$. Це приводить нас до остаточного стандартного формулювання задачі мінімізації в методі опорних векторів.

Знайти параметри $\vec{w}$ і *b*, що задовольняють наступним умовам. (7.6)

- Величина $\dfrac{1}{2}\vec{w}^T\vec{w}$ досягає мінімуму.
- При усіх $(\vec{x}_i, y_i) \in \mathbb{D}$ виконується нерівність $y_i\left(\vec{w}^T\vec{x}_i + b\right) \geq 1$.

[2] Нагадаємо, що $|\vec{w}| = \sqrt{\vec{w}^T\vec{w}}$.

Отже, необхідно мінімізувати квадратичну функцію при лінійних обмеженнях. Задача *квадратичної оптимізації* — це добре вивчена математична задача оптимізації, для рішення якої розроблено багато алгоритмів. У принципі, реалізувати метод опорних векторів можна за допомогою стандартних бібліотек квадратичного програмування, але останнім часом з'явилося багато робіт, що пропонують спеціалізовані методи квадратичного програмування для реалізації методу опорних векторів. У результаті розроблені більш складні, але більш швидкі і масштабовані бібліотеки, що широко використовуються для побудови моделей. Опис цих алгоритмів не входить у нашу задачу.

Однак для розуміння сутності методу опорних векторів корисно привести наступну інформацію про розв'язування поставленої оптимізаційної задачі. Для того щоб знайти розв'язок, необхідно сформулювати двоїсту задачу, у якій з кожним обмеженням $y_i\left(\vec{w}^T\vec{x}_i+b\right)\geq 1$ прямої задачі зв'язаний відповідний множник Лагранжа α_i.

Знайти $\alpha_1, \alpha_2, ..., \alpha_N$, за яких величина $\sum\alpha_i-\frac{1}{2}\sum_i\sum_j\alpha_i\alpha_j y_i y_j\vec{x}_i^T\vec{x}_j$ досягає максимуму (7.7)

- $\sum_i\alpha_i y_i=0$,
- $\alpha_i\geq 0$ при усіх $1\leq i\leq N$.

Розв'язок цієї задачі має наступний вид.

$$\vec{w}=\sum\alpha_i y_i\vec{x}_i \tag{7.8}$$

$b=y_k-\vec{w}^T\vec{x}_k$ при будь-яких $\vec{x}_k$, таких що $\alpha_k\neq 0$

У цьому розв'язку більшість параметрів α_i дорівнює нулю. Кожне ненульове значення α_i означає, що відповідний вектор $\vec{x}_i$ є опорним. Таким чином, функція класифікації має наступний вид.

$$f\left(\vec{x}\right)=sign\left(\sum_i\alpha_i y_i\vec{x}_i^T\vec{x}+b\right) \tag{7.9}$$

Вираз, яких необхідно максимізувати в двоїстій задачі, як і функція класифікації, містить скалярний добуток пар точок ($\vec{x}$ і $\vec{x}_i$ чи $\vec{x}_i$ і $\vec{x}_j$), і це єдиний спосіб використання даних. Значення цього факту стане ясним пізніше.

Отже, ми починаємо з навчальної множини. Ця сукупність даних однозначно визначає найкращу поділяючу гіперплощину, що є результатом розв'язування задачі квадратичної оптимізації. Якщо нова точка $\vec{x}$ підлягає класифікації, то функція класифікації $f\left(\vec{x}\right)$, визначена або рівністю (7.1), або рівністю (7.9), обчислює проекцію цієї точки на нормаль гіперплощини. Знак цієї функції визначає клас, якому належить точка. Якщо точка лежить усередині зазору класифікатора (чи іншої довірчої смуги *t*, що ми установимо для помилок класифікації), то класифікатор відповідає "не знаю" і не вибирає жодний з класів. Значення функції $f\left(\vec{x}\right)$ можна перетворити в імовірність класифікації; як правило, для цього підбирають придатний сигмоїд. Крім того, зазор є постійним, тому, якщо модель включає розмірності різного походження, може знадобитися ретельне масштабування. Однак це не проблема, якщо наші документи (точки) лежать на одиничній гіперсфері.

Приклад 7.1. Розглянемо процес створення класифікатора по методу опорних векторів на основі (дуже маленької) множини даних, показаної на рис. 7.4. З геометричної точки зору ваговий вектор, що максимізує зазор, паралельний найкоротшій лінії, що з'єднує точки з двох класів, тобто лінії, що проходить через точки (1, 1) і (2, 3), що дає нам ваговий вектор (1, 2). Оптимальна поділяюча поверхня ортогональна цієї лінії і перетинає її посередині. Отже, вона проходить через точку (1,5; 2). Отже, роздільна поверхня по методу опорних векторів має вид

$$y = x_1 + 2x_2 - 5{,}5.$$

З алгебраїчної точки зору ми повинні мінімізувати функцію $\left|\vec{w}\right|$ за умови $y_i\left(\vec{w}^T\vec{x}_i+b\right)\geq 1$. Це відбудеться, якщо дане обмеження перетвориться в рівність на двох опорних векторах. Крім того, відомо, що розв'язок має вид $\vec{w}=(a,2a)$ при деякому *a*. Отже, необхідно вирішити систему рівнянь

$$a + 2a + b = -1$$
$$2a + 6a + b = 1.$$

Отже, a = 2/5 і b = –11/5. Таким чином, оптимальна півплощина визначається параметрами $\vec{w}$ =(2/5; 4/5) і b = -11/5.

Зазор ρ дорівнює $2/|\vec{w}| = 2/\left(4/25 + 15/25\right) = 2/\left(2\sqrt{5}/5\right) = \sqrt{5}$. Ця відповідь можна підтвердити геометрично, проаналізувавши рис. 7.4.

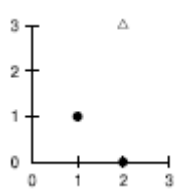

Рис.7.4. Трьохточкова навчальна множина даних для методу опорних векторів

Вправа 7.1. Яку мінімальну кількість опорних векторів може мати набір даних, що містить екземпляри кожного класу?

Вправа 7.2. Можливість використання ядер у методі опорних векторів (розділ 7.2.3) виникає завдяки тому, що функцію класифікації можна записати у виді (7.9), де для великих задач майже всі числа α_i дорівнюють нулю. Покажіть, як можна записати функцію класифікації в цьому виді для даних із вправи 7.1. Інакше кажучи, запишіть f як функцію, у яку входять точки даних, і єдиною змінною є вектор $\vec{x}$.

Вправа 7.3. Інсталлируйте який-небудь програмний пакет, що реалізує метод опорних векторів, наприклад SVMlight (`http://svmlight.joachims.org/`), і побудуйте класифікатор для набору даних, описаного в прикладі 7.1. Переконаєтеся, що програма приводить до тих же самим результатам, що зазначені в тексті. Файл може мати наступний вид, типовий для таких пакетів.

```
+1 1:2 2:3
-1 1:2 2:0
-1 1:1 2:1
```

Команда на навчання в пакеті SVMlight має наступний вид.

```
svm_learn -c 1 -a alphas.dat train.dat model.dat
```

Опція `-в 1` необхідна для того, щоб відключити використання фіктивних перемінних, котрі ми обговоримо в розділі 7.2.1. Переконаєтеся, що норма вагового вектора погодиться з результатами, отриманими при виконанні вправи 7.1. Перевірте файл `alphas.dat`, що містить значення α_i, і переконаєтеся, що вони збігаються з відповідями, отриманими при виконанні вправи 7.2.

7.2. Розширення моделі опорних векторів

7.2.1. Класифікація з м'яким зазором

У задачах дуже великої розмірності, типових для класифікації текстів, дані іноді допускають лінійний поділ. Однак у загальному випадку це припущення не виконується, а якщо і виконується, та перевага все рівно варто віддати рішенню, що краще розділяє основну масу даних, ігноруючи невелику кількість незвичайних шумових документів.

Якщо навчальна множина $\mathbb{D}$ не є лінійно роздільною, то звичайно при побудові широкого розділового зазору допускається декілька помилок (деякі точки — викиди чи шумові екземпляри — можуть лежати усередині зазору чи на невірній стороні). За кожен невірно класифікований екземпляр накладається штраф, що залежить від того, наскільки сильно порушуються умови (7.5), накладені на зазор. Для цього в задачу вводяться *фіктивні змінні* ξ_i. Ненульове значення змінної ξ_i дозволяє вектору $\vec{x}$ порушувати вимоги, що висуваються до зазору, але за це накладається штраф, пропорційний величині ξ_i (рис. 7.5).

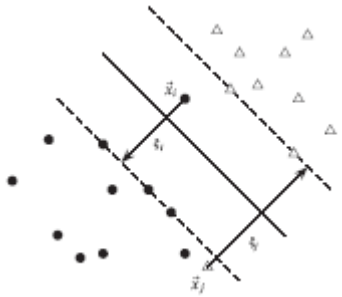

Рис. 7.5.. Класифікація із широким зазором і фіктивними перемінними

Формулювання задачі оптимізації в методі опорних векторів з фіктивними перемінними виглядає так.

Знайти параметри $\vec{w}$, b і $\xi_i \geq 0$, при яких (7.10)

- функція $\frac{1}{2}\vec{w}^T\vec{w} + C\sum_i \xi_i$ досягає мінімуму і
- при усіх $\left\{\left(\vec{x}_i, y_i\right)\right\}$ виконується нерівність $y_i\left(\vec{w}^T\vec{x}_i + b\right) \geq 1 - \xi_i$.

Рішення оптимізаційної задачі носить компромісний характер: воно встановлює баланс між шириною зазору і кількістю точок, що довелося б перемістити, для того щоб забезпечити цю ширину. Установивши фіктивне значення $\xi_i > 0$, ширину зазору для точки $\vec{x}$ можна зробити менше одиниці, але за це доведеться заплатити штраф $C\xi_i$. Сума величин ξ_i визначає верхню границю кількості помилок при навчанні. Метод опорних векторів з м'яким зазором (soft-margin SVM) мінімізує кількість помилок при навчанні за рахунок ширини зазору. Параметр C називається *параметром регуляризації*. Він дозволяє керувати перенавчанням: якщо параметр C стає великим, то небажано ігнорувати дані за рахунок зменшення геометричного зазору; якщо параметр C невеликий, то за допомогою фіктивних перемінних можна легко врахувати деякі точки й одержати зазор, що моделює основну масу даних.

Двоїста задача класифікації з м'яким зазором формулюється так.

Знайти параметри $\alpha_1, \alpha_2, ..., \alpha_N$, що максимізують функцію (7.11)

$\sum \alpha_i - \frac{1}{2}\sum_i\sum_j \alpha_i\alpha_j y_i y_j \vec{x}_i^T\vec{x}_j$ за умов

- $\sum_i \alpha_i y_i = 0$
- $0 \leq \alpha_i \leq C$ при усіх $1 \leq i \leq N$.

У двоїстій задачі не з'являються ані фіктивні перемінні ξ_i, ані множник Лагранжа для них. У ній залишається тільки константа C, що обмежує можливу величину множника Лагранжа для опорних векторів. як і колись, опорними векторами є точки $\vec{x}_i$ з ненульовими значеннями α_i. Рішення двоїстої задачі має наступний вид.

$$\vec{w} = \sum \alpha y_i \vec{x}_i \tag{7.12}$$

$$b = y_k\left(1 - \xi_k\right) - \vec{w}^T\vec{x}_k \text{ для } k = \arg\max_k \alpha_k$$

Як і колись, вектор $\vec{w}$ у явному виді для класифікації не потрібний. Класифікацію можна здійснити за допомогою обчислення скалярного добутку, як у формулі (7.9).

Як правило, опорні вектори складають невелику частину навчальної множини. Однак якщо задача не має властивості лінійної роздільності чи зазор малий, то кожна невірно класифікована точка чи точка, що лежить усередині зазору, буде мати ненульове значення α_i. Якщо множина таких точок стає великою, то в нелінійному випадку це може виявитися основним фактором зниження уповільнення методу опорних векторів на етапі тестування.

Складність навчання і тестування за допомогою лінійного методу опорних векторів показана в табл. 7.1. Час навчання в методі опорних векторів в основному визначається часом розв'язування відповідної задачі квадратичного програмування, тому теоретична й емпірична складність залежить від способу розв'язування цієї задачі. Вважається, що часова складність стандартного рішення задачі квадратичного програмування пропорційна кубу обсягу набору даних. Усі недавні роботи з методу опорних векторів спрямовані на зниження

цієї складності, причому досить часто це відбувається за рахунок того, що точний розв'язок заміняється наближеним. Як правило емпірична складність цих методів складає $O(|\mathbb{D}|^{1,7})$. Понадлінійна складність традиційних алгоритмів опорних векторів утрудняє і навіть іноді унеможливлює їхнє застосування до великих наборів навчальних даних. Альтернативні алгоритми опорних векторів, складність яких лінійно залежить від кількості навчальних вибірок, погано масштабуються для великої кількості ознак, що є характерним для задач класифікації текстів. Однак нові багатообіцяючі алгоритми навчання, засновані на методі площин, що відтинають, лінійно залежать від кількості навчальних прикладів і кількості ненульових ознак у них. І все ж реальна швидкість квадратичної оптимізації набагато нижче швидкості простого підрахунку термінів у наївної байєсівської моделі. Заміна лінійного методу опорних векторів нелінійним, як буде показано в наступному розділі, звичайно приводить до підвищення часової складності навчання в $|\mathbb{D}|$ разів (оскільки необхідно обчислювати скалярний добуток елементів навчальної множини), що зовсім неприйнятно. На практиці частіше дешевше створити ознаки більш високого порядку і провести навчання за допомогою лінійного методу опорних векторів.

Таблиця 7.1. Складність навчання і тестування різних класифікаторів, включаючи метод опорних векторів. Час навчання — це час, що метод навчання затрачає на настроювання класифікатора за допомогою множини $\mathbb{D}$, а час тестування — це час, що класифікатор затрачає на класифікацію одного документа. Для методу опорних векторів передбачається, що класифікація по декількох класах виробляється за допомогою сукупності $|\mathbb{C}|$ бінарних (один клас — всі інші) класифікаторів. L_{ave} — це середня кількість лексем на документ, а M_{ave} — середній розмір лексикона документа (кількість ненульових ознак). L_a і M_a — кількість лексем і різних термінів у тестовому документі відповідно

Класифікатор	Вид	Метод	Часова складність
Наївний байєсівський	Навчання		$\theta\,(\|\mathbb{D}\|L_{ave} + \|\mathbb{C}\|\|V\|)$
Наївний байєсівський	Тестування		$\theta\,(\|\mathbb{C}\|M_a)$
Роккіо	Навчання		$\theta\,(\|\mathbb{D}\|L_{ave} + \|\mathbb{C}\|\|V\|)$
Роккіо	Тестування		$\theta(\|\mathbb{C}\|M_a)$
kNN	Навчання	Попередня обробка	$\theta(\|\mathbb{D}\|L_{ave})$
kNN	Тестування	Попередня обробка	$\theta(\|\mathbb{D}\|M_{ave}M_a)$
kNN	Навчання	Без попередньої обробки	$\theta(1)$
kNN	Тестування	Без попередньої обробки	$\theta\,(\|\mathbb{D}\|L_{ave}M_{ave})$
Метод опорних векторів	Навчання	Звичайний	$O(\|\mathbb{C}\|\|\mathbb{D}\|^3 M_{ave})$ $\approx$ $O(\|\mathbb{C}\|\|\mathbb{D}\|^{1,7} M_{ave})$, емпірично
Метод опорних векторів	Навчання	Площини, що відтинають	$O(\|\mathbb{C}\|\|\mathbb{D}\|L_{ave}M_{ave})$
Метод опорних векторів	Тестування		$O(\|\mathbb{C}\|M_a)$

7.2.2. Метод опорних векторів з декількома класами

Стандартний метод опорних векторів призначений для класифікації по двох класах. Традиційно для класифікації по декількох класах за допомогою методу опорних векторів використовується один з методів, описаних у лекції 7. Зокрема, на практиці частіше створюються $|\mathbb{C}|$ класифікаторів, що працюють за принципом "один проти інших" (іноді цей принцип називається "один проти всіх" (One-Versus-All — OVA)), а потім вибирається клас, на якому тестовий документ відстоїть далі усього від поділяючої поверхні. Інша стратегія полягає в побудові сукупності класифікаторів, що працюють за принципом "один проти одного", а потім вибирається клас, запропонований більшістю класифікаторів. Незважаючи на те що ця процедура передбачає створення $|\mathbb{C}|(|\mathbb{C}|-1)/2$ класифікаторів, час на їхнє навчання на ділі може знизитися, оскільки навчальна множина для кожного класифікатора набагато менше.

Однак усі ці стратегії не дуже елегантні. Набагато краще розробити класифікатор для декількох класів, побудувавши бінарний класифікатор по векторі ознак $\Phi(\vec{x}, y)$, побудованому по парам, що складається з вхідних ознак і відповідного класу. На етапі тестування класифікатор вибирає клас $y = \arg\max_{y'} \vec{w}^T \Phi(\vec{x}, y')$. Зазором на етапі навчання є різниця між значеннями, що відповідають правильному і найближчий неправильному класам, тому задача квадратичного програмування містить наступну умову: $\forall i \; \forall y \neq y_i \; \vec{w}^T \Phi(\vec{x}_i, y_i) - \vec{w}^T \Phi(\vec{x}_i, y) \geq 1 - \xi_i$. За допомогою цього загального методу можна сформулювати задачу багатокласової класифікації для різних лінійних класифікаторів. Крім того, це приклад простого узагальнення класифікації, у якому класи являють собою не просту множина незалежних категоріальних міток, а можуть бути довільно структурованими об'єктами з взаємними залежностями. Такі варіанти методу опорних векторів називаються *структурними*.

7.2.3. Нелінійний метод опорних векторів

Дотепер ми розглядали випадки, коли набори даних допускали лінійний поділ (можливо, з невеликими виключеннями чи шумами). А що ж робити, якщо сукупність даних не дозволяє застосовувати лінійні класифікатори? Розглянемо одномірний випадок. Дані, приведені у верхній частині рис. 7.6, легко розпізнаються лінійним класифікатором, а дані в середній частині — ні. Для того щоб розпізнати дані в середній частині, потрібно виділити інтервал. Один зі способів рішення цієї проблеми полягає у відображенні даних у простір більш високої розмірності з наступним застосуванням лінійного класифікатора в цьому просторі. Наприклад, у нижній частині рис. 7.6 показано, що лінійний класифікатор легко розпізнає дані, якщо для відображення даних у двовимірну площину використовується квадратична функція (іншим варіантом є полярні координати). Основна ідея полягає у відображенні вихідного простору ознак у простір ознак більш високої розмірності, у якому навчальна множина виявляється лінійно роздільним. Зрозуміло, при цьому бажано зберегти релевантну розмірність відношень між точками даних, так, щоб отриманий класифікатор узагальнював вихідні дані.

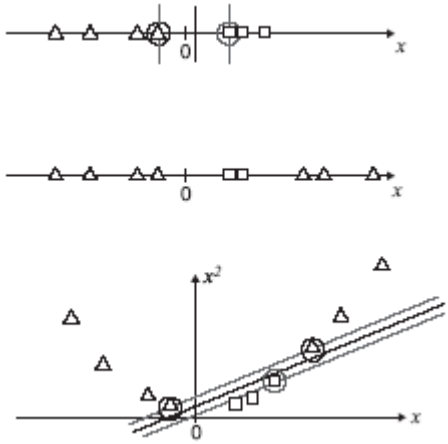

Рис.7.6. Проекція даних, що не допускають лінійного поділу, у простір більш високої розмірності, у якому лінійна роздільність існує

Метод опорних векторів, як і ряд інших лінійних класифікаторів, дозволяє легко й ефективно здійснювати таке відображення даних у простір більш високої розмірності. Цей прийом називається *переходом до ядра* (kernel trick). Лінійний класифікатор, побудований по методу опорних векторів, заснований на обчисленні скалярного добутку між векторами, що відповідають даним. Уведемо наступне позначення: $K(\vec{x}_i, \vec{x}_j) = \vec{x}_i \vec{x}_j^T$. У такому випадку уже відомий нам класифікатор можна переписати інакше.

$$f(\vec{x}) = sign\left(\sum_i \alpha_i y_i K(\vec{x}_i, \vec{x}) + b\right) \tag{7.13}$$

Припустимо тепер, що ми вирішили відобразити кожну точку в простір більш високої розмірності, використовуючи функцію $\Phi : \vec{x} \mapsto \phi(\vec{x})$. У такому випадку скалярний добуток перетворюється в добуток $\phi(\vec{x}_i)^T \phi(\vec{x}_j)$. Якщо виявиться, що цей скалярний добуток (який являє собою дійсне число) можна обчислити відносно просто й ефективно по вихідних точках, то відображення $\vec{x} \mapsto \phi(\vec{x})$ насправді здійснювати

необов'язково. Замість цього можна просто обчислити величину $K\left(\vec{x}_i,\vec{x}_j\right)=\boldsymbol{\phi}\left(\vec{x}_i\right)^T\boldsymbol{\phi}\left(\vec{x}_j\right)$, а потім використовувати значення функції у формулі (7.13). *Ядро K* — це функція, що відповідає скалярному добутку в деякому розширеному просторі ознак.

Приклад 7.2. **Квадратичне ядро на площині.** Для двовимірних векторів $\vec{u}=\begin{pmatrix}u_1 & u_2\end{pmatrix}$ і $\vec{v}=\begin{pmatrix}v_1 & v_2\end{pmatrix}$ розглянемо функцію $K\left(\vec{u},\vec{v}\right)=\left(1+\vec{u}^T\vec{v}\right)^2$. Покажемо, що вона є ядром, тобто $K\left(\vec{u},\vec{v}\right)=\boldsymbol{\phi}\left(\vec{u}\right)^T\boldsymbol{\phi}\left(\vec{v}\right)$ при якомусь $\boldsymbol{\phi}$. Розглянемо вектор $\boldsymbol{\phi}(u)=\begin{pmatrix}1 & u_1^2 & \sqrt{2}u_1u_2 & u_2^2 & \sqrt{2}u_1 & \sqrt{2}u_2\end{pmatrix}$. У такому випадку

$$\begin{aligned}K\left(\vec{u},\vec{v}\right)&=\left(1+\vec{u}^T\vec{v}\right)^2=\\&=1+u_1^2v_1^2+2u_1v_1u_2v_2+u_2^2v_2^2+2u_1v_1+2u_2v_2=\\&=\begin{pmatrix}1 & u_1^2 & \sqrt{2}u_1u_2 & u_2^2 & \sqrt{2}u_1 & \sqrt{2}u_2\end{pmatrix}^T=\\&=\begin{pmatrix}1 & v_1^2 & \sqrt{2}v_1v_2 & v_2^2 & \sqrt{2}v_1 & \sqrt{2}v_2\end{pmatrix}=\\&=\boldsymbol{\phi}\left(\vec{u}\right)^T\boldsymbol{\phi}\left(\vec{v}\right).\end{aligned}\tag{7.14}$$

Використовуючи термінологію функціонального аналізу, поставимо запитання: "Які функції можуть бути *ядрами* (kernels)?" Ядра іноді точніше називати *ядрами Мерсера*, тому що вони повинні задовольняти умові Мерсера: при будь-якій функції $g\left(\vec{x}\right)$, такий що інтеграл $\int g^2\left(\vec{x}\right)d\vec{x}$ є скінченним, повинне виконуватися умову

$$\int K\left(\vec{x},\vec{z}\right)g\left(\vec{x}\right)g\left(\vec{z}\right)d\vec{x}d\vec{z}\geq 0\ .\tag{7.15}$$

Ядро *K* повинне бути неперервним, симетричним, а також мати позитивно визначену матрицю Грама. Ці умови гарантують, що існує відображення у відтворююче ядро гільбертова простору, тобто простір, скалярний добуток у який збігається зі значенням функції *K*. Якщо ядро не задовольняє умові Мерсера, то відповідна задача квадратичного програмування може не мати розв'язку. Для того щоб краще зрозуміти ці проблеми, рекомендуем звернутися до книг, перерахованим у розділі 7.5.

Найбільш розповсюдженими сімействами ядер є поліноміальні ядра і функції радіального базису. Поліноміальні ядра мають вид $K\left(\vec{x},\vec{z}\right)=\left(1+\vec{x}^T\vec{z}\right)^d$. При *d* = 1 ядро є лінійним. Саме з таким ядром ми працювали до цього розділу (константа 1 просто змінює поріг). При *d* = 2 виникає квадратичне ядро, що також широко використовується. Квадратичне ядро описане в прикладі 7.2.

Найбільш розповсюдженою формою функції радіального базису є функція щільності гаусова розподілу

$$K\left(\vec{x},\vec{z}\right)=e^{-\left(\vec{x}-\vec{z}\right)/\left(2\sigma^2\right)}\ .\tag{7.16}$$

Функція радіального базису еквівалентна відображенню даних у нескінченовимірний гільбертовий простір, тому функцію радіального базису неможливо проілюструвати так само конкретно, як квадратичне ядро.

1. Маннинг К.Д., Рагхаван П., Шютце Х. Введение в информационный поиск.— М.: Изд-во "Вильямс", 2011.