Вирішальні дерева

7.1. Вирішальні дерева

Вирішальні дерева - це сімейство алгоритмів, яке дуже відрізняється від лінійних моделей, але в той же час відіграє важливу роль в машинному навчанні.

7.1.1. Лінійні моделі (огляд)

До цього моменту вивчали лінійні моделі. До особливостей лінійних моделей належить таке:

- Лінійні моделі швидко навчаються. У випадку із середньоквадратичною помилкою для вектора терезів навіть є аналітичне рішення. Також легко застосовувати для лінійних моделей градієнтний спуск.
- При цьому лінійні моделі можуть відновлювати лише прості залежності через обмежену кількість параметрів (ступеня свободи).
- У той же час лінійні моделі можна використовувати для відновлення нелінійних залежностей за рахунок переходу до простору, що спрямовує, що є досить складною операцією.

Окремо варто відзначити, що лінійні моделі не відображають особливості процесу ухвалення рішень у людей. Насправді, коли людина хоче зрозуміти ту чи іншу річ, вона задаватиме послідовність із простих питань, які в результаті приведуть її до якоїсь відповіді.

7.1.2. Вирішальні дерева (приклад 1)

Щоб зрозуміти принцип роботи вирішальних дерев, корисно розглянути такий спрощений приклад.

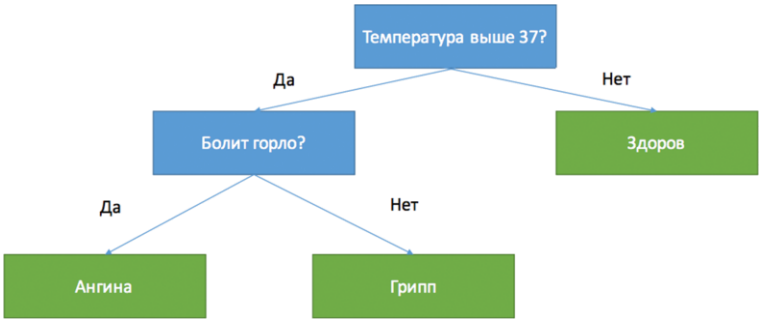

Рис. 7.1

Потрібно провести медичну діагностику. Лікар, який проводить цю діагностику, знає лише 2 захворювання – ангіна та грип. Тому спочатку він питає, яка температура у пацієнта. Якщо вона менше 37 градусів, він робить висновок, що пацієнт здоровий, в іншому випадку — переходить до наступного питання, а саме, запитує, чи болить у пацієнта горло. Якщо воно болить, лікар ставить діагноз ангіна, інакше грип.

Творці курсу не рекомендують серйозно ставитись до запропонованого методу діагностики захворювань.

7.1.3. Вирішальні дерева (приклад 2)

Інший приклад — для відомого завдання визначення того, чи виживе чи не виживе той чи інший пасажир Титаніка. Завдання дуже добре вирішується наступним вирішальним деревом:

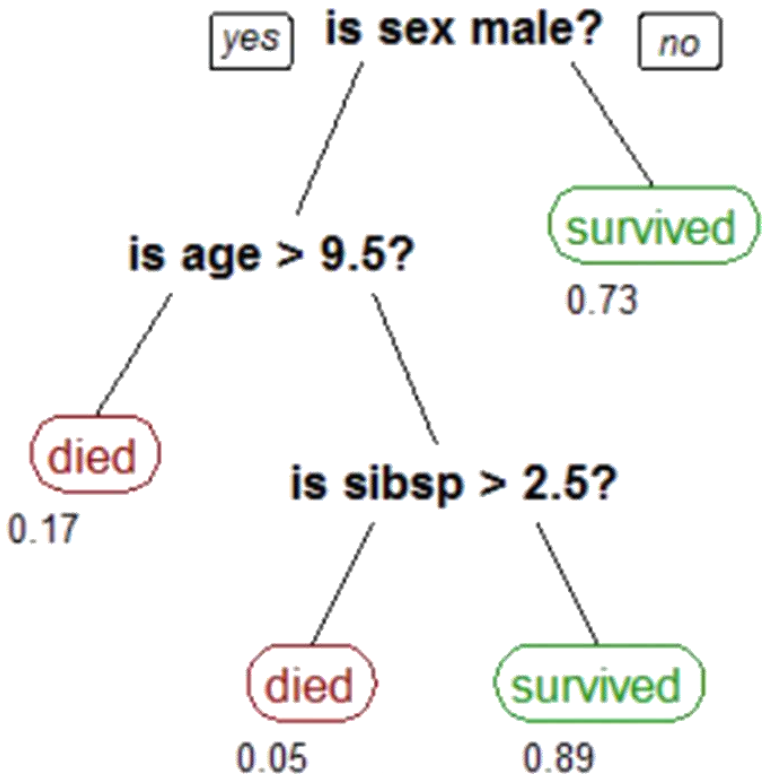

Рис. 7.2

Насамперед питається пів пасажира. Якщо це жінка, то вирішальне дерево відразу заявляє, що вона виживає, і ця відповідь вірна у 73% випадків, і так далі.

7.1.4. Вирішальні дерева

Отже, було розглянуто два приклади вирішальних дерев, які були бінарними деревами, в кожній внутрішній вершині записано умову, а в кожному аркуші дерева — прогноз. Строго кажучи, не обов'язково вирішальне дерево має бути бінарним, але зазвичай використовуються саме бінарні.

Умови у внутрішніх вершинах вибираються дуже простими. Найчастіший варіант - перевірити, чи лежить значення деякої ознаки xj лівіше, ніж заданий поріг t:

$$[x^j \le t].$$

Це дуже проста умова, яка залежить від однієї ознаки, але її достатньо, щоб вирішувати багато складних завдань.

Прогноз у листі є дійсним числом, якщо вирішується завдання регресії. Якщо ж вирішується завдання класифікації, то як прогноз виступає або клас, або розподіл ймовірностей класів.

7.1.5. Вирішальні дерева у задачі класифікації

Нехай вирішується завдання класифікації з двома ознаками та трьома класами.

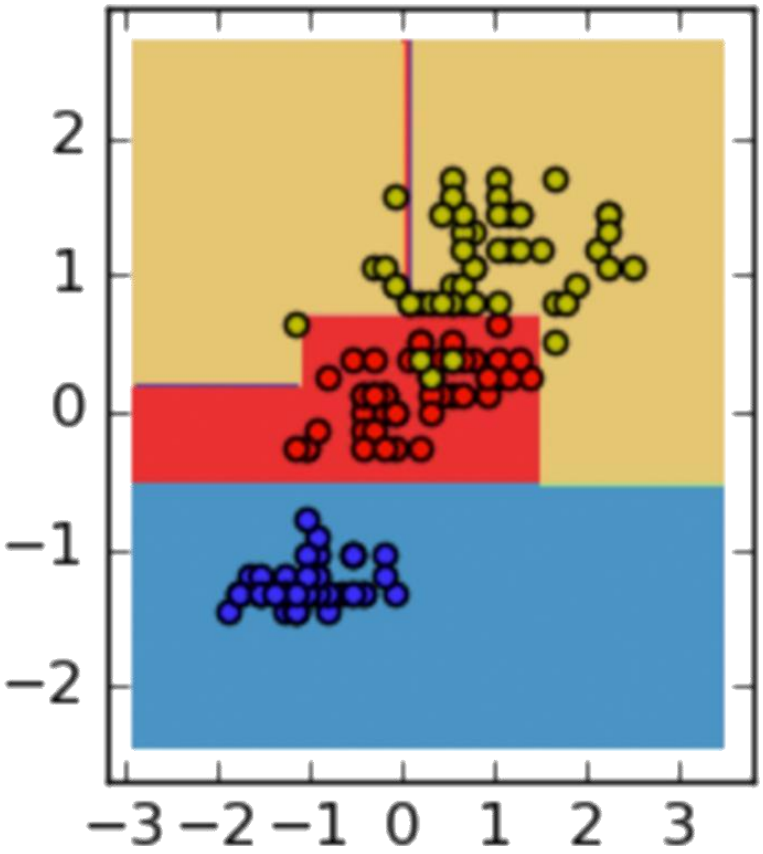

Рис. 7.3: Використання вирішальних дерев у задачах класифікації

Видно, що вирішальне дерево може дуже непогано відокремити кожен клас від решти. Видно, що розділяюча поверхня кожного класу кусково-постійна, і при цьому кожна сторона поверхні паралельна осі координат, так як кожна умова порівнює значення одно ознаки з порогом.

У той же час вирішальне дерево цілком може перенавчитися: його можна зробити настільки глибоким, що кожен лист вирішального дерева буде відповідати рівно одному об'єкту навчальної вибірки. У цьому випадку, якщо записати в кожному аркуші відповідь відповідного об'єкта, на вибірці виходить нульова помилка. Дерево виходить явно перенавченим. Ось приклад такого дерева:

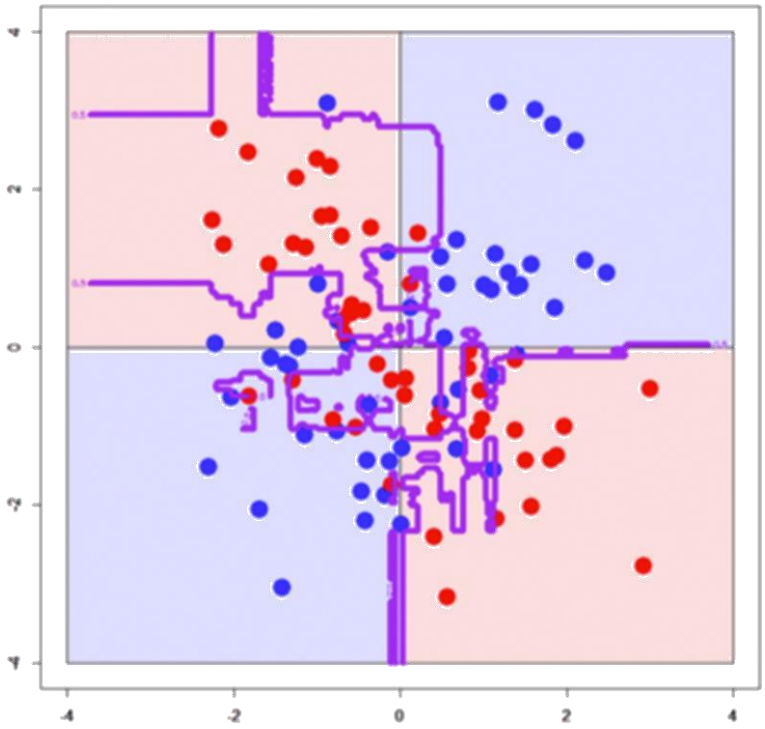

Рис. 7.4: Перенавчене вирішальне дерево

Це дерево ідеально відокремило синій від червоного класу, але поверхня, що розділяє, вийшла шалено складною - видно, що цей алгоритм перенавчився і від нього не буде ніякої користі на тестовій вибірці.

7.1.6. Вирішальні дерева в задачі регресії

Нехай вирішується завдання регресії з одним ознакою, яким потрібно відновити значення цільової змінної. Не дуже глибоке дерево відновлює залежність приблизно так:

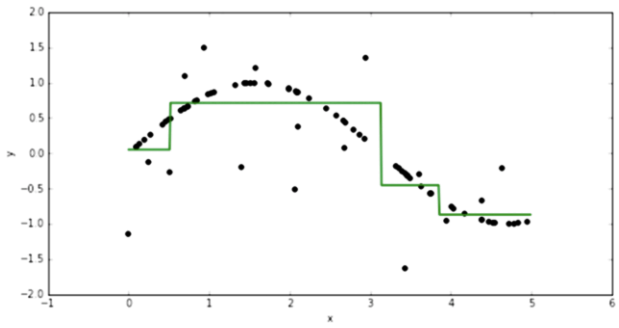

Рис. 7.5: Використання вирішальних дерев у задачах регресії

Відновлена залежність буде шматково-постійною, але загалом матиме непогану якість.
При збільшенні глибини дерева функція, що вийшла, матиме наступний вигляд:

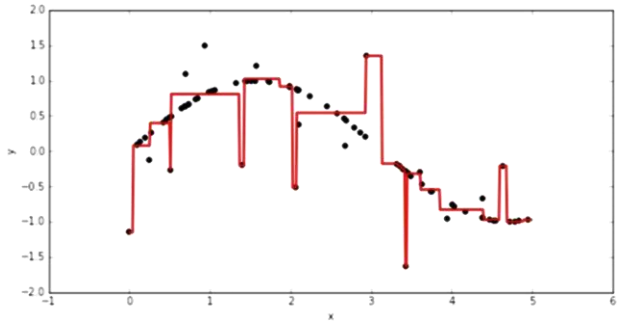

Рис. 7.6: Перенавчене вирішальне дерево

Видно, що дерево підігналося під викиди і його якість вже буде не такою гарною. Дерево перенавчилося через те, що його глибина занадто велика.

7.2. Навчання вирішальних дерев

У цьому розділі буде розглянуто питання, як будувати вирішальні дерева та як навчати їх за конкретною вибіркою.

7.2.1. Перенавчання дерев

У попередньому розділі було показано, що вирішальні дерева легко перевчаються. У тому числі можна побудувати дерево, у якого кожен лист відповідатиме одному об'єкту навчальної вибірки..

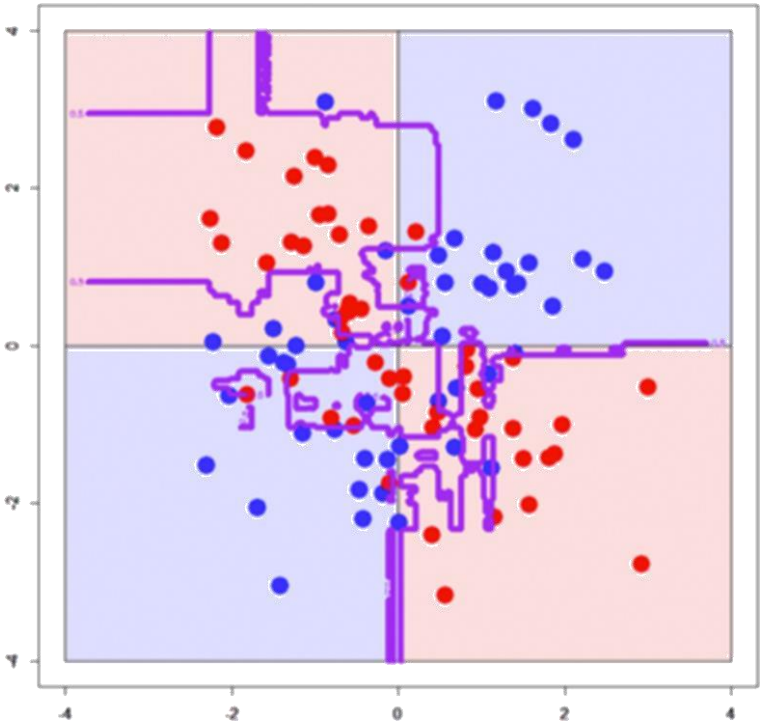

Рис. 7.7: Перенавчене вирішальне дерево

Це дерево будує дуже складну розділяючу поверхню і, очевидно, перенавчено.

Оскільки завжди можна побудувати таке дерево, яке не помиляється на навчальній вибірці і буде перенавченим, має сенс шукати мінімальне (наприклад, з мінімальним числом листя) дерево яке має нульову помилку. Але, на жаль, завдання відшукання такого дерева – NP-повне, тобто його неможливо вирішити за розумний час.

7.2.2. Жадібний спосіб побудови

У машинному навчанні застосовується жадібний спосіб побудови вирішального дерева від кореня до листя. Спочатку вибирається корінь, який розбиває вибірку на дві. Потім розбивається кожен із нащадків цього кореня і таке інше. Дерево розгалужується до тих пір, поки цього не буде достатньо.

Залишається уточнити спосіб розбиття кожного нащадка. Як було сказано раніше, як умова в кожній вершині дерева, що будується, буде використовуватися найпростіша умова: значення однієї з ознак порівнюватиметься з деяким порогом.

Нехай у вершину m потрапило безліч Xm об'єктів із навчальної вибірки. Параметри в умові *xj ≤ t+ будуть вибрані так, щоб мінімізувати цей критерій помилки Q(Xm,j,t), який залежить від цих параметрів: $Q(X_m,j,t) \to \min_{j,t}$.

Параметри j та t можна підбирати перебором. Справді, ознак кінцеве число, та якщо з усіх можливих значень порога t можна лише ті, у яких виходять різні розбиття. Можна показати, що таких значень параметра t стільки, скільки різних значень xj на навчальній вибірці.

Після того, як параметри були обрані, безліч Xm об'єктів з навчальної вибірки розбивається на дві множини

$$X_\ell = \left\{x \in X_m | [x^j \le t]\right\}, \qquad X_r = \left\{x \in X_m | [x^j > t]\right\},$$

кожне з яких відповідає своїй дочірній вершині.

Запропоновану процедуру можна продовжити для кожної з дочірніх вершин: у цьому випадку дерево все більше і більше поглиблюватиметься. Такий процес рано чи пізно має зупинитися, і чергова дочірня вершина буде оголошена листком, а чи не розділена навпіл. Цей момент визначається критерієм зупинки. Існує багато різних варіантів критерію зупинки:

- Якщо у вершину потрапив лише один об'єкт навчальної вибірки чи всі об'єкти належать одному класу (у завданнях класифікації), далі розбивати немає сенсу.
- Можна також зупиняти розбиття, якщо глибина дерева досягла певного значення.

Можливі критерії зупинки обговорюватимуться пізніше у цьому уроці.

Якщо якась вершина не була поділена, а була оголошена аркушем, потрібно визначити прогноз, який утримуватиметься в даному аркуші. У цей лист потрапила деяка підвибірка Xm вихідної навчальної вибірки і потрібно вибрати такий прогноз, який буде оптимальним для цієї підвиборки.

У задачі регресії, якщо функціонал — середньоквадратична помилка, оптимально давати середню відповідь на цю вибірку:

$$a_m = \frac{1}{|X_m|} \sum_{i \in X_m} y_i.$$

У задачі класифікації оптимально повертати той клас, який найбільш популярний серед об'єктів Xm:

$$a_m = \operatorname{argmax}_{y \in Y} \sum_{i \in X_m} [y_i = y].$$

Якщо потрібно вказати ймовірності класів, їх можна вказати як частку об'єктів різних класів у Xm:.

$$a_{mk} = \frac{1}{|X_m|} \sum_{i \in X_m} [y_i = k]$$

7.3. Критерії інформативності

У цьому розділі йтиметься про критерії інформативності, за допомогою яких можна вибирати оптимальне розбиття при побудові вирішального дерева.

7.3.1. Вибір критерію помилки Критерій

помилки записується так:

$$Q(X_m, j, t) = \frac{|X_\ell|}{|X_m|} H(X_\ell) + \frac{|X_r|}{|X_m|} H(X_r)$$ і

складається з двох доданків, кожне з яких відповідає своєму листу.

Функція H(X) називається критерієм інформативності: її значення має бути тим менше, чим менше розкид відповідей у X.

У разі регресії розкид відповідей — це дисперсія, тому критерій інформативності у завданнях регресії записується так:

$$H(X) = \frac{1}{|X|} \sum_{i \in X} (y_i - \bar{y}(X))^2, \qquad \bar{y} = \frac{1}{|X|} \sum_{i \in X} y_i$$.

7.3.2. Критерій інформативності Джині

Сформулювати критерій інформативності завдання класифікації дещо складніше. Нехай pk - частка об'єктів класу k у вибірці X:

$$p_k = \frac{1}{X} \sum_{i \in X} [y_i = k]$$.

Критерій інформативності Джіні формулюється в термінах pk:

$$H(X) = \sum_{k=1}^{K} p_k (1 - p_k)$$.

Всі складові в сумі невід'ємні, тому критерій Джіні також негативний. Його оптимум досягається тільки в тому випадку, коли всі об'єкти X відносяться до одного класу.

Одна з інтерпретацій умов Джіні - це можливість помилки випадкового класифікатора. Класифікатор влаштований таким чином, що можливість видати клас k дорівнює pk.

7.3.3. Ентропійний критерій інформативності

Ще один критерій інформативності – ентропійний критерій:

$$H(X) = -\sum_{k=1}^{K} p_k \ln p_k$$.

У цьому вся виразі вважається, що 0ln0 = 0.

Ентропійний критерій, як і критерій Джині, невід'ємний, а його оптимум також досягається тільки в тому випадку, коли всі об'єкти X відносяться до одного класу.

Ентропійний критерій має цікавий фізичний зміст. Він у тому, що показує, наскільки розподіл класів X відрізняється від виродженого. Ентропія у разі виродженого розподілу дорівнює 0: такий розподіл характеризується мінімальним можливим ступенем несподіванки. Навпаки, рівномірний розподіл найнесподіваніший, і йому відповідає максимальна ентропія.

7.4. Критерій зупинки та стрижка дерев

У цьому розділі мова йтиме про способи боротьби з перенавчанням дерев, а саме про умови зупинки та стрижку дерев.

7.4.1. Критерій зупинки

Критерій зупинки використовується, щоб ухвалити рішення: розбивати вершину далі або зробити листовий.

Найгірший випадок вирішального дерева - таке, в якому кожен лист відповідає своєму об'єкту навчальної вибірки. У цьому випадку дерево буде максимально перенавченим і не узагальнюватиме інформацію, отриману з навчальної вибірки. Грамотно підібраний критерій зупинки дозволяє боротися з перенавчанням.

Найпростіший критерій зупинки перевіряє, чи всі об'єкти у вершині відносяться до одного класу. Однак такий критерій зупинки може бути використаний тільки у разі простих вибірок, так як для складних він зупиниться лише тоді, коли в кожному аркуші залишиться приблизно по одному об'єкту.

Набагато стійкіший і корисніший критерій перевіряє, скільки об'єктів виявилося на вершині, і розбиття триває, якщо це число більше, ніж деяке обране n. Відповідно, якщо до вершини потрапило ≤ n об'єктів, вона стає листовою. Параметр n слід підбирати.

Випадок n = 1 є найгіршим випадком, описаним вище. При цьому вибирати n потрібно так, щоб за n об'єктами, які потрапили у вершину, можна було стійко побудувати прогноз. Існує рекомендація, що потрібно брати рівним 5.

Ще один критерій, набагато грубіший, полягає в обмеженні на глибину дерева. Цей критерій добре зарекомендував себе при побудові композицій, коли багато вирішальних дерев об'єднують в один складний алгоритм. Про це йтиметься пізніше.

7.4.2. Стрижка дерев

Існує й інший підхід до боротьби з перенавчанням дерев – стрижка. Він полягає в тому, що спочатку будується вирішальне дерево максимальної складності та глибини, доти, поки в кожній вершині не виявиться по 1 об'єкту навчальної вибірки.

Після цього починається «стрижка», тобто видалення листя у цьому дереві за певним критерієм. Наприклад, можна стригти доти, доки покращується якість деякої відкладеної вибірки.

Існує думка, і це підкріплено багатьма експериментами, що стрижка працює набагато краще, ніж прості критерії, про які йшлося раніше. Але стрижка - дуже ресурсомістка процедура, оскільки, наприклад, може знадобитися обчислення якості дерева на певній валідаційній вибірці на кожному кроці.

Насправді, самі по собі дерева на сьогоднішній день майже не використовуються, вони бувають потрібні лише для побудови композиції та об'єднання великої кількості дерев в один алгоритм. У випадку з композиціями такі складні підходи до боротьби з перенавчанням вже не потрібні, тому що досить простих критеріїв зупинки, обмеження на глибину дерева або кількість об'єктів в листі.

7.5. Вирішальні дерева та категоріальні ознаки

У цьому розділі буде розказано, як використовувати категоріальні ознаки у вирішальних деревах.

До цього моменту використовувалася така умова у вершині кожного дерева:

*xj ≤ t+.

Очевидно, що таку умову можна записувати лише для речових чи бінарних ознак. **7.5.1.**

N-арні дерева

Підхід, який дозволяє включити категоріальні ознаки в дерева, полягає в тому, щоб будувати nарні дерева, тобто такі дерева, що з кожної вершини можуть виходити до n ребер. Нехай необхідно розбити певну вершину за деякою ознакою. Якщо ця ознака — речова чи бінарна, то все ще можна використовувати просту умову з порогом t, тому інтерес представляє саме випадок категоріальної ознаки.

Якщо xj — категоріальна ознака, яка може набувати значень

$$\{c_1,\ldots,c_n\},$$

можна розбити вершину на n вершин таким чином, що в і дочірню вершину йдуть об'єкти з xj = ci. Критерій помилки такого розбиття будується за аналогією з випадком бінарного дерева. Оскільки Xn розбивається на n частин, а чи не на дві, у виразі буде n доданків:

$$Q(X_m, j) = \sum_{i=1}^{n} \frac{|X_i|}{|X_m|} H(X_i) \to \min_j .$$

Таким чином, якщо вершину m потрібно розбити, розглядаються всі можливі речові, бінарні та категоріальні ознаки. Для речових і бінарних ознак вважається Q(Xm,j,t), а n-арных — оскільки написано вище. Розбиття вершини відбуватиметься за тією ознакою, для якої значення критерію помилки буде мінімальним.

Важливе зауваження полягає в тому, що при розподілі вершини за категоріальною ознакою виходить більше дочірніх вершин, на яких, ймовірно, буде досягатися більш висока якість і більш низьке значення критерію інформативності, тому, швидше за все, при такому підході перевага майже завжди віддаватиметься розбиття за категоріальними ознаками з великою кількістю можливих значень. В результаті виходить багато листя в дереві, що майже гарантовано може призвести до перенавчання.

Однак, це не завжди так. Якщо вибірки настільки великі, що навіть після розбиття за категоріальною ознакою в кожному піддереві залишатиметься багато об'єктів, то таке дерево буде непогано працювати, оскільки воно відновлюватиме складні залежності, і при цьому не перевчуватиметься при використанні належного критерію зупинки.

7.5.2. Бінарні дерева з розбиттям безлічі значень

Інший підхід дозволяє не переходити до n-арних дерев та продовжувати працювати з бінарними деревами. Нехай також необхідно зробити розбиття вершини m, а категоріальна ознака xj може набувати значень C = ,c1,...,cn-.

Для цього спочатку необхідно розбити безліч значень категоріальної ознаки на два підмножини, що не перетинаються:

$$C = C_1 \cup C_2, \qquad C_1 \cap C_2 = \emptyset .$$

Після того, як таке розбиття збудовано, умова в даній вершині виглядатиме просто:

$$[x^j \in C_1]$$

Ця умова перевіряє, у яке з підмножин потрапляє значення ознаки на даний момент цьому об'єкті. Головним питанням залишається те, як потрібно розбивати безліч C.

Повна кількість можливих розбиття множини на дві підмножини - 2n. На щастя, є хитрість, яка дозволяє уникнути повного перебору і, більше того, працювати з категоричною ознакою як із речовим. І тому можливі значення категориального ознаки сортуються спеціальним чином c(1),...,c(n) і замінюються натуральні числа 1,...,n. Після цього з даними ознакою слід вже працювати як з речовим, а значення порога t визначатиме поділ множини C на два підмножини.

Сортувати значення категоріальної ознаки у разі завдання бінарної класифікації слід за таким принципом:

$$\frac{\sum_{i \in X_m} [x_i^j = c_{(1)}][y_i = +1]}{\sum_{i \in X_m} [x_i^j = c_{(1)}]} \le \ \dots \ \le \frac{\sum_{i \in X_m} [x_i^j = c_{(n)}][y_i = +1]}{\sum_{i \in X_m} [x_i^j = c_{(n)}]}$$

Фактично значення категоріальної ознаки сортуються за зростанням частки об'єктів +1 класу серед об'єктів вибірки Xn з відповідним значенням цієї ознаки.

Для задачі регресії сортування відбувається схожим чином, але обчислюється не частка об'єктів позитивного класу, а середня відповідь по всіх об'єктах, у яких значення категоріальної ознаки дорівнює c:

$$\frac{\sum_{i \in X_m} [x_i^j = c_{(1)}] y_i}{\sum_{i \in X_m} [x_i^j = c_{(1)}]} \le \ \dots \ \le \frac{\sum_{i \in X_m} [x_i^j = c_{(n)}] y_i}{\sum_{i \in X_m} [x_i^j = c_{(n)}]} .$$

Головна особливість такого підходу полягає в тому, що отриманий результат є повністю еквівалентним результату, який можна було б отримати в результаті повного перебору. Ця умова працює для критерію Джині, MSE та ентропійного критерію.