Лекція 3. Метод *k* найближчих сусідів [1]

На відміну від методу Роккіо, *метод k найближчих сусідів* (k nearest neighbor), чи *kNN-класифікація*, визначає поділяючі межі локально. У варіанті 1NN кожна ознака відноситься до визначеного класу в залежності від інформації про його найближчого сусіда. У варіанті kNN кожна ознака відноситься до переважного класу найближчих сусідів, де k-параметр методу. В основі методу kNN лежить факт, что відповідно до гіпотези компактності ми очікуємо, что тестова ознака d буде мати таку ж мітку, як і навчальні ознаки в локальній області, що оточує ознаку d.

Поділяючі межі в методі 1NN являють собою суміжні сегменти *діаграми Вороного* (Voronoi tesselation), показаної на рис. 3.1. Діаграма Вороного для множини об'єктів розділяє простір на осередки, що складаються з точок, що ближче до даного об'єкта, ніж до інших. У нашому випадку об'єктами є ознаки. Діаграма Вороного розділяє площину на $|\mathbb{D}|$ опуклих багатокутників, кожний з який містить відповідну ознаку (і не містить інших), як показано на рис. 3.1, на якому опуклий багатокутник є опуклою областю двовимірного простору, обмеженою лініями.

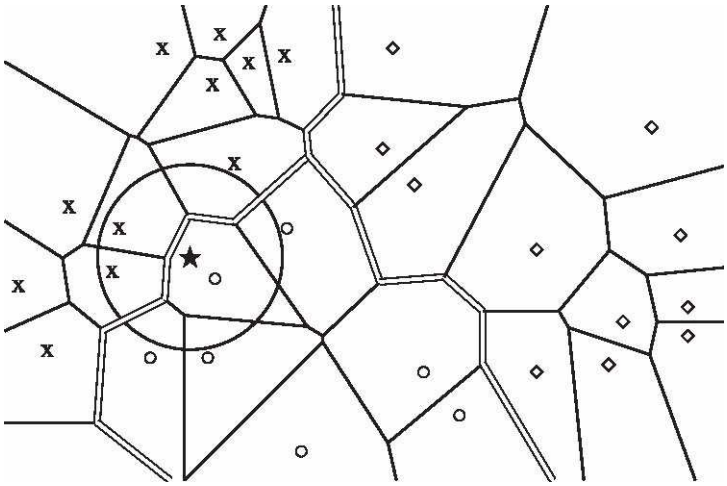

Рис. 3.1. Діаграма Вороного і поділяючі межі (подвійні лінії) у методі 1NN. Показано три класи: хрестики, кружечки і ромбики

Для довільного параметра $k \in \mathbb{N}$ у методі kNN розглянемо область простору, для якої множина k найближчих сусідів залишається однаковою. Ця область також являє собою опуклий багатокутник, а простір виявляється розділеним на опуклі багатокутники, усередині кожного з якою множина k найближчих сусідів є інваріантною (вправу 2.11).

Метод 1NN не дуже стійкий. Класифікація кожної текстової ознаки залежить від класу, до якого належить окрема навчальна ознака, що може мати невірну мітку чи взагалі бути нетиповим. Метод kNN при $k > 1$ є більш стійким. Він приписує ознаки до переважаючого класу по k найближчих сусідах, випадковим образом розриваючи зв'язку між ними.

Існує імовірнісний варіант методу kNN. Можна оцінити імовірність того, що ознака належить класу c, як частку k найближчих сусідів у класі c. На рис. 3.1 наведений приклад класифікації при $k = 3$. Оцінки імовірностей того, що ознака, відзначена зірочкою, належить трьом класам, такі: P(клас кружечків|зірочка) = 1/3, P(клас хрестиків|звездочка) = 2/3, P(клас ромбиків|зірочка) = 0. Оцінки методу 3NN (P(клас кружочків|зірочка) = 1/3) і методу 1NN (P(клас кружечків|зірочка) = 1) відрізняються, тому метод 3NN віддає перевагу класу хрестиків, а метод 1NN — класу кружечків.

Параметр k у методі kNN часто вибирається на підставі досвіду чи знань про розв'язувану задачу класифікації. Бажано, щоб параметр k був непарним, щоб зменшити імовірність «нічиєї». Найчастіше вибираються значення $k = 3$ і $k = 5$, але використовуються і великі значення, між 50 і 100. Як альтернативу параметр k можна вибрати так, щоб він гарантував найкращі результати на відкладеній частині навчального множини.

Можна також приписати ваги “голосам” k найближчих сусідів, використовуючи їх косинусну міру подібності. У цій схемі ранги класів обчислюються так.

$$ранг(c,d) = \sum_{d' \in S_k} I_c(d') \cos\left(\vec{v}(d'), \vec{v}(d)\right)$$

Тут S_k — множина k найближчих сусідів ознаки d' і $I_c(d') = 1$ тоді і только тоді, коли ознака d' належить класу c і $I_c(d') = 0$ — у супротивному випадку. Ознака приписується до класу з найбільшим рангом. Зважування за допомогою міри подібності часто виявляється більш точним, ніж просте голосування. Наприклад, якщо два класи мають однакову кількість сусідів, то вибирається клас з більш близькими сусідами.

Алгоритм kNN Навчання (з попередньою обробкою) і тестування по методу kNN. Величина p_j — це оцінка імовірності $P(c_j|S_k) = P(c_j|d)$, а c_j — множина всіх ознак у класі c_j**)**

```
Train-kNN(ℂ, 𝔻)
1   𝔻′ ⟵ Preprocess(𝔻)
2   k ⟵ Select-k{ℂ,𝔻′}
3   return 𝔻′, k
```

```
Apply-kNN(ℂ,𝔻′, k, d)
1   S_k ⟵ ComputeNearestNeighbors(𝔻′, k, d)
2   for each c_j ∈ ℂ
3   do p_j ⟵ |S_k ∩ c_j|/k
4   return arg max_j p_j
```

Приклад 3.1. Відстані між тестовою ознакою і чотирма навчальними ознаками в табл. 3.1 рівні $\left|\vec{d}_1-\vec{d}_5\right|=\left|\vec{d}_2-\vec{d}_5\right|=\left|\vec{d}_3-\vec{d}_5\right|\approx 1,41$ і $\left|\vec{d}_4-\vec{d}_5\right|=0,0$. Отже, найближчим сусідом ознаки d_5 є ознака d_4, і метод 1NN припише ознаку d_5 до класу ознаки d_4, тобто до класу $\bar{c}$.

Часова складність і оптимальність методу kNN

Оцінки часової складності методу kNN наведені в табл. 3.1. Метод kNN досить сильно відрізняється від інших алгоритмів класифікації. Навчання методу kNN зводиться до простого визначення параметра k і попередній обробці ознак. Фактично, якщо параметр k обраний заздалегідь, а попередня обробка ознак не передбачена, у методі kNN взагалі відсутня фаза навчання. На практиці попередня обробка, наприклад розбивка на лексеми (tokenization), є обов'язковою. Її доцільно провести один раз на етапі навчання, а не повторювати щораз при класифікації нової ознаки.

Таблиця 3.1. Оцінки часової складності навчання і тестування для класифікації kNN (M_{ave} — середній розмір лексикона колекції)

kNN з попередньою обробкою навчального множини	
Навчання	$\Theta(\|\mathbb{D}\|L_{ave})$
Тестування	$\Theta(L_a+\|\mathbb{D}\|M_{ave}M_a)=\Theta(\|\mathbb{D}\|M_{ave}M_a)$
kNN без попередньої обробки навчального множини	
Навчання	$\Theta(1)$
Тестування	$\Theta(L_a+\|\mathbb{D}\|L_{ave}M_a)=\Theta(\|\mathbb{D}\|L_{ave}M_a)$

Часова складність тестування в методі kNN дорівнює $\Theta(|\mathbb{D}|M_{ave}M_a)$. Вона лінійно залежить від розміру навчальної множини, оскільки при класифікації необхідно обчислити відстань від тестової ознаки до кожної ознаки з навчального множини. Тривалість тестування не залежить від кількості класів J. Отже, при великій кількості класів J метод kNN має потенційні переваги.

У класифікації по методу kNN не виробляється оцінка жодного параметру, як у методі Роккіо (центроїди) або в наївному байєсівському методі (апріорні й умовні імовірності). У методі kNN всі екземпляри з навчальної множини просто запам'ятовуються, а потім порівнюються з тестовою ознакою. З цієї причини метод kNN також називають *навчанням на основі запам'ятовування* (memory-based learning) і *навчанням на основі запам'ятовування прецедентів* (instance-based learning). Для машинного навчання бажано мати якнайбільше навчальних даних. Однак у методі kNN великі навчальні множини приводять до значного зниження ефективності класифікації.

Чи можна провести тестування по методу kNN швидше, ніж $\Theta(|\mathbb{D}|M_{ave}M_a)$, або, ігноруючи довжину ознак, ефективніше, ніж $\Theta(|\mathbb{D}|)$. При невеликих размерностях M існують швидкі алгоритми kNN. При великих размірностях M розроблені наближені методи, що гарантують визначений діапазон помилок при заданому рівні ефективності. Ці наближені методи не пройшли широкої апробації на задачах класифікації текстів, тому поки неясно, чи можуть вони досягти ефективності, більшої, чем $\Theta(|\mathbb{D}|)$, без значної втрати точності.

Існує подібність між задачею пошуку найближчого сусіда тестової ознаки і задачею пошуку по довільному запиті, метою якого є ознаки, що мають найбільшу схожість із запитом. Фактично ці дві проблеми являють собою різні варианти задачі про k найближчих сусідів і відрізняються лише відносною щільністю вектора тестової ознаки (сотні ненульових елементів у методі kNN) і розрідженістю вектора запиту (як правило, у задачі пошуку по довільному запиті кількість ненульових елементів менше десяти). Для забезпечення ефективності довільного пошуку використовується інвертований індекс. Чи можна створити ефективний метод kNN, використовуючи аналогічний інвертований індекс?

Інвертований індекс обмежує пошук только тими ознаками, що мають хоча б один загальний термін із запитом. Таким чином, у контексті методу kNN інвертований індекс виявиться ефективним, якщо тестова ознака не має загальних термінів з великою кількістю навчальних ознак. Чи це так, залежить від конкретної задачі класифікації. Якщо ознаки довгі, а список стоп-слів не використовується, то економія часу буде незначною. Однак при коротких ознаках і довгому списку стоп-слів інвертований індекс може скоротити середній час тестування в десять і більш разів.

Час пошуку при використанні інвертованого індексу залежить від довжини списку словопозицій термінів із запиту. Довжина списків словопозицій сублінійно зростає при збільшенні розміру колекції, оскільки лексикон збільшується відповідно до закону Хіпса (Heaps' law): якщо імовірність появи одних термінів зростає, то імовірність появи інших повинна убувати. Однак більшість нових термінів не є розповсюдженими. Отже, складність пошуку при використанні інвертованого індексу залишається рівною $\Theta(T)$, і якщо середня довжина ознаки згодом не змінюється, то $\Theta(T) = \Theta(|\mathbb{D}|)$.

Якість класифікації методу kNN близька до якості найбільш точних методів навчання в класифікації текстів (табл. 15.2). Мірою якості методу навчання є *рівень байєсівської помилки* (Bayes error rate), тобто середній рівень помилок класифікаторів, навчених за допомогою цього методу для рішення конкретної задачі. Метод kNN не оптимальний для проблем з ненульовою байєсівською помилкою, тобто для проблем, при рішенні яких навіть найкращий із усіх можливих класифікаторів має ненульову помилку класифікації. Помилка методу 1NN асимптотично (при навчальній множині, що збільшується,) обмежена подвоєним рівнем байєсівської помилки. Інакше кажучи, якщо оптимальний класифікатор має рівень помилок, рівний x, то асимптотичний рівень помилок методу 1NN дорівнює $2x$. Це пояснюється наявністю шуму. Джерелами шуму є два компоненти методу kNN: тестова ознака і найближча навчальна ознака. Ці джерела є аддитивними, тому сумарна помилка методу 1NN удвічі перевищує оптимальний рівень помилки. Для задач з нульовим байєсівським рівнем помилки рівень помилки методу 1NN при збільшенні розміру навчальної множини прямує до нуля.

Вправа 3.1. Поясніть, чому метод kNN з мультимодальними класами працює краще, ніж метод Роккіо.

Рекомендації

1. Якщо ознаки є упорядкованими, обчислюється євклідова відстань.

$$d(x, y) = \sqrt{(x_i - y_i)^2}$$, де n – кількість ознак.

2. Якщо ознаки не можна упорядкувати, використовується метрика ізольованих точок.

$$d(x, y) = \begin{cases} 0, x = y, \\ 1, x \neq y. \end{cases}$$

3. Для того щоб уникнути впливу різних масштабів ознак, застосовують нормалізацію. Існує два види нормалізації: мінімаксна і за допомогою стандартного відхилення.

$$\overline{x}_i = \frac{x_i - \min\limits_{i=1,n} x_i}{\max\limits_{i=1,n} x_i - \min\limits_{i=1,n} x_i}$$ (мінімаксна нормалізація)

$$\overline{x}_i = \frac{x_i - \overline{x}}{s}$$ (нормалізація за допомогою стандартного відхилення),

де $\overline{x}$ — середнє значення, s — стандартне відхилення.

Переваги

- Алгоритм стійкий до аномальних викидів, тому що імовірність улучення такого запису в число k-найближчих сусідів мала. Якщо ж це відбулося, то вплив на голосування (особливо зважене) (при k>2) також, швидше за все, буде незначним, і, отже, малим буде і вплив на підсумок класифікації.
- Програмна реалізація алгоритму відносно проста.
- Результат роботи алгоритму легко піддається інтерпретації. Експертам у різних областях цілком зрозуміла логіка роботи алгоритму, заснована на пошуку схожих об'єктів.
- Можливість модифікації алгоритму, шляхом використання найбільш придатних функцій сполучення і метрик дозволяє підбудувати алгоритм під конкретну задачу.

Недоліки

- Набір даних, використовуваний для алгоритму, повинний бути репрезентативним.
- Модель не можна "відокремити" від даних: для класифікації нового прикладу потрібно використовувати всі приклади. Ця особливість сильно обмежує використання алгоритму

Література

1. Маннинг К.Д., Рагхаван П., Шютце Х. Введение в информационный поиск.— М.: Изд-во «Вильямс», 2011.
2. Воронцов К. Машинное обучение (курс лекций). http://www.machinelearning.ru/wiki/index.php?title=Машинное_обучение_(курс_лекций%2C_К.В.Воронцов)