1. Задачі класифікації

Базові алгоритми класифікації:
1. Байєсів класифікатор.
2. Метод к-найближчих сусідів.
3. Логістична регресія.

Задача класифікації (з учителем) це задача прогнозування факторного або категоріального відгуку. Як і в задачі регресії ми матимемо вектор Y довжини N. та матрицю регерсорів $X \in \mathbb{R}^{N\times p}$. У задачах класифікації Y- це не числова змінна!

2. Наївний баєсів класифікатор.

Нехай $y \in \{1, \dots, K\}$.

$$P\{y_j = k \mid x_j\} = P\{y_j = k\}P\{x_j|y_j = k\}/C, \tag{1}$$

де C - деяка загальна константа, що не залежить від k. Тепер нам потрібно оцінити апріорний розподіл y та умовні розподіли x відносно y. Для того, щоб оцінити апріорний розподіл y викоритовують звичайний вектор частот y. Оцінити $P\{x|y\}$ як правило складно з технчних причин. Тому, для оцінки цієї ймовірності використовують так зване наївне припущення:

$$P\{x = (x^1, \dots, x^p)|y = k\} = \prod_{j=1}^{p} P\{x^j|y = k\}.$$

$$\hat{y} = argmax_{k\in\{1,\dots,K\}} P\{y = k|x\} = argmax_{k\in\{1,\dots,K\}} P\{y = k\}P\{x|y = k\}.$$

2.1. **Приклад.** . Класифікація спаму. $y \in \{S, H\}$. Що буде x в такій моделі? Розглянемо деякий словник, розміру p - набір з p впорядкованих по алфавіту слів. Тоді вектор $x_j = (x_j^1, \dots, x_j^p)$ - це вектор частот. Тобто x_j^l - це частота l-того слова у повідомленні j. Типово, вектор x_j - це вектор з багатьма нулями, і лише деякі значення ненульові. Таким чином $P\{x^l|S\}$ - оцінюється частотою входження слова з індексом l у спамові повідомлення, а $P\{x^l|H\}$ - аналогічна частота входження у не спамові повідомлення. Тоді, для тестового повідомлення $y, x = (x^1, \dots, x^p)$, ймовірність $P\{y = S\}$ або $P\{y = H\}$ оцінюється за формулою (1).

Однією з переваг НБК є те, що x - не обов'язково повинен бути числовим.

3. Метод к-найближчих сусідів.

Розглядаються лише числові регресори. Кожне спостереження зображується у вигляді точки в $\mathbb{R}^p$. Тобто наш тренувальний набір буде зображено у вигляді N точок в $\mathbb{R}^p$. До кожної точки з тренувального набору приписана мітка, або її клас.

Далі, для тестової точки $x = (x^1, \dots, x^p)$, шукаємо k точок із тренувальної вибірки які найближчі до x в деякій метриці (як правило евклідовій). Після цього, кожна із k-точок "голосує" відповідно до свого класу. Той клас який набрав найбільше "голосів" приписується x. Алгоритм $k - nn$ непогано працює при $k = 1$! Як правило вибирається стандартна метрика в $\mathbb{R}^p$. Іноді вибирають метрику Махалонобіса:

$$d(x, y) = \sqrt{(x - y)^T A(x - y)}.$$

4. Логістична регресія

Логістична регресія - це алгоритм класифікації, тобто вектор Y - це факторний вектор, і припустимо для почтаку, що $y_i \in \{0, 1\}$. Логістична випливає зі спроби застосувати лінійну регресію до задач класифікації. Як може виглядати спроба застосувати лінійну регресію до задачі класифікації?

$$Y = X\beta + \varepsilon$$

цей підхід має цілий ряд недоліків:
1. По-перше дискретну множину значень $\{0, 1\}$ ми наближаємо прямою, що приймає значення від $-\infty$ до $+\infty$.
2. По друге пряма значення між 0 та 1.
3. По-третє прямою дуже складно наблизити дискретну набір точок з множини $\{0, 1\}$.

Як можна змінити цю модель? По-перше можна спробувати оцінювати $P\{y = 0\}$ або $P\{y = 1\}$. Потрібно щось зробити з $X\beta$, щоб він не приймав занадто великих значень і від'ємних значень.

Для цього потрібно застосувати покоординатно функцію

$$\sigma(z) = \frac{1}{1 + e^{-z}}.$$

Якщо ми застосуємо цю функцію, то дуже малі (або від'ємні) значення $x_i\beta = \sum_{k=1}^{p} x_i^k \beta_k$ стануть близькими до нуля, а дуже великі значення близькими до 1. Насправді можна вибирати будь-яку функцію яка має подібні властивості, наприклад функція $\Phi(z)$ - функція стандартного нормального розподілу.

В подальшому, замість β будемо писати $\omega \in \mathbb{R}^p$. Тоді наша модель матиме вигляд:

$$Y = \sigma(X\omega) + \varepsilon,$$

тобто

$$y_i = \sigma(\sum_{k=1}^{p} x_i^k \omega_k) + \varepsilon_i.$$

Зауважимо, що в такій моделі y - є випадковою величиною, яка приймає два значення 0 та 1. Якщо припустити, що $E\varepsilon_i = 0$, то:

$$EY = \sigma(X\omega).$$

Але $Ey = 0P\{y = 0\} + 1P\{y = 1\} = P\{y = 1\}$. Таким чином отримали:

$$P_{x,\omega}\{y = 1\} = \sigma(x\omega),\ P_{x,\omega}\{y = 0\} = 1 - \sigma(x\omega).$$

Зауважимо, що часто в літературі пишуть $P\{y = 1|x, \omega\}$ - але це не умовна ймовірність, бо ні x ні ω не є випадковими!. Тоді вираз для $P_{x,\omega}\{y = i\}$ це можна переписати:

$$p_{x,\omega}(y) = \sigma^y(x\omega)(1 - \sigma(x\omega))^{1-y},\ x \in \mathbb{R}^{1\times p}, \omega \in \mathbb{R}^{p\times 1}$$

Запишемо фунцію правдоподібності (нагадування $Y \in \mathbb{R}^{N\times 1}$- вектор тренувальних відгуків, $X \in \mathbb{R}^{N\times p}$ - матриця регресорів):

$$L(\omega) = p_{X,\omega}(Y) = \prod_{i=1}^{N} p_{x_i,\omega}(y_i) = \prod_{i=1}^{N} \sigma(x_i\omega)^{y_i}(1 - \sigma(x_i\omega))^{1-y_i},$$

розглянемо логарифм:

$$l(\omega) = \ln L(\omega) = \sum_{i=1}^{N} y_i \ln \sigma(x_i\omega) + (1 - y_i) \ln(1 - \sigma(x_i\omega)),$$

якщо $\sigma(x_i\omega)$ це наш прогноз, то доцільно позначити його $\hat{y}_i = \sigma(x_i\omega)$, тоді цільова функція матиме вигляд:

$$l(\omega) = \sum_{i=1}^{N} y_i \ln \hat{y}_i + (1 - y_i) \ln(1 - \hat{y}_i) \to \max.$$

Оскільки $l(\omega)$ при неправильній класифікації приймає великі від'ємні значення, а при правильній значення близькі до нуля, то розглядаються функцію:

$$J(\omega) = -\frac{1}{N} l(\omega) = -\frac{1}{N} \sum_{i=1}^{N} y_i \ln \hat{y}_i + (1 - y_i) \ln(1 - \hat{y}_i) \to \min.$$

ця функція називається ентропією. Мінімум даної задачі не виражається якось простою формулою, а як правило знаходиться чисельними методами, як метод Ньютона-Раффсона або градієнтний спуск.

Відступ. Метод січних Ньютона полягає у тому, що при мінімізації функції $J(\omega)$ за певних умов регулярності, має місце збіжність до **локального мінімуму** послідовності

$$\omega^{(n+1)} = \omega^{(n)} - J(\omega)/J'(\omega),$$

якщо ω-скаляр.

Метод градієнтного спуску заснований на тому, що вектор градієнту спрямований у напрямку (локального) максимуму. Тому для функції: $F\colon \mathbb{R}^n \to \mathbb{R}$, послідовність $x^{(n+1)} = x^{(n)} + \alpha \nabla F(x^{(n)})$ збігається до локального максимум, за певних умов регулярності, а якщо потрібно мінімізувати F, то

$$x^{(n+1)} = x^{(n)} - \alpha \nabla F(x^{(n)}),$$

збігатиметься до локального мінімуму. Оптимізація методом градієнтного спуску полягає в наступному:
1. Обирається деяке $x^{(0)}$
2. Обчислюється градієнт в точці $x^{(0)}$
3. Обчислюється $x^{(1)}$
4. Повторюються кроки 1-3 з $x^{(1)}$ замість $x^{(0)}$
5. і так далі поки зміна цільової функції не буде меншою за деяке наперед задане ε, тобто $|F(x^{(n+1)}) - F(x^{(n)})| < \varepsilon$.
α - параметр, який підганається під конкретну задачу.

Продовження.
Покажемо, як обчислити ω методом градієнтного спуску. Для цього потрібно обчислити градієнт $J(\omega)$. Зауважимо, що функція $J(\omega) = J(\hat{y}(\omega))$.

$$\frac{\partial J}{\partial \omega} = \frac{\partial J}{\partial \hat{y}} \frac{\partial \hat{y}}{\omega}.$$

$\hat{y} = \hat{y}(\omega_1, \ldots, \omega_p)\colon \mathbb{R}^p \to\colon \mathbb{R}^N$, $\hat{y}_i(\omega) = \sigma(x_i\omega)$, $J(\hat{y})\colon \mathbb{R}^N \to \mathbb{R}$. тоді $J(\omega)\colon \mathbb{R}^p \to \mathbb{R}$.

$$\frac{\partial J}{\partial \hat{y}} \in \mathbb{R}^{1\times N},\ \frac{\partial \hat{y}}{\partial \omega} \in \mathbb{R}^{N\times p}.$$

Обчислимо $\frac{\partial \hat{y}}{\partial \omega}$ зауваживши, що $\sigma' = \sigma(1-\sigma)$.

$$\frac{\partial \hat{y}}{\partial \omega} = \begin{pmatrix} \frac{\partial \hat{y}_1(\omega)}{\omega_1} & \dots & \frac{\partial \hat{y}_1(\omega)}{\omega_p} \\ \vdots & \dots & \vdots \\ \frac{\partial \hat{y}_N(\omega)}{\omega_1} & \dots & \frac{\partial \hat{y}_N(\omega)}{\omega_p} \end{pmatrix} \tag{2}$$

Оскільки $\hat{y}_i(\omega) = \sigma(x_i\omega)$, то $\frac{\partial \hat{y}_i(\sigma)}{\partial \omega_k} = \sigma(x_i\omega)(1-\sigma(x_i\omega))x_i^k = \hat{y}_i(1-\hat{y}_i)x_i^k$ Тоді матриця $\frac{\partial \hat{y}}{\partial \omega}$ матиме вигляд:

$$\frac{\partial \hat{y}}{\partial \omega} = UX, \tag{3}$$

$$U = \begin{pmatrix} \hat{y}_1(1-\hat{y}_1) & 0 & \dots & 0 \\ \vdots & & \dots & \vdots \\ 0 & 0 & 0 & \hat{y}_N(1-\hat{y}_N) \end{pmatrix} \tag{4}$$

Легко показати, що

$$\frac{\partial J}{\partial \hat{y}} = -\frac{1}{N}(Y-\hat{Y})^T U^{-1} = (\hat{Y}-Y)^T U^{-1}$$

Обчилимо,

$$\frac{\partial J}{\partial \hat{y}_i} = -\frac{1}{N}\left(\frac{y_i}{\hat{y}_i} - \frac{1-y_i}{1-\hat{y}_i}\right) = \frac{1}{N}\frac{\hat{y}_i - y_i}{\hat{y}_i(1-\hat{y}_i)}$$

Виходить така формула:

$$\frac{\partial J}{\partial \omega} = \frac{1}{N}(\hat{Y}-Y)^T U^{-1} UX = \frac{1}{N}(\hat{Y}-Y)X$$

Тепер методом градієнтного спуску можна обчислити послідовність $\omega^{(n)} \in \mathbb{R}^p$ що збігається до локального мінімума $J(\omega)$.

$$\omega^{(n+1)} = \omega^{(n)} + \frac{1}{N}\alpha((Y-\hat{Y})^T X)^T = \omega^{(n)} + \frac{1}{N}\alpha X^T(Y-\hat{Y})$$