Машинно обучение

Лабораторно упражнение 6

Логаритмична регресия. Регуляризация

Целта на упражнението е да се реализира логаритмична регресия в класификационна задача както и да се демонстрира използването на регуляризация за получаване на адекватна хипотеза (модел).

I. Логаритмична регресия без регуляризация

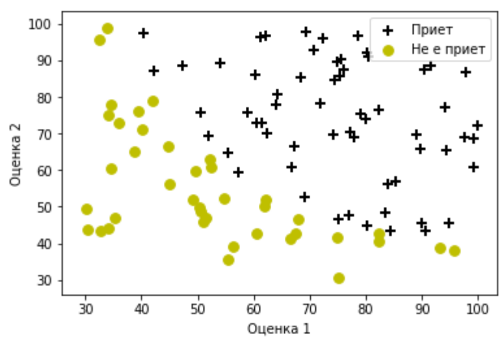
Като пример ще използваме следната задача:

Разполагаме с данни за оценки на кандидат-студенти от два изпита (резултатите са в брой точки). Решението дали студентът ще бъде приет в университета или не се взема от комисия. В обучителните данни за всеки студент имаме резултат 1 (приет в университета) или 0 (не е приет). Ще построим хипотеза, която предсказва дали студент ще бъде приет на базата на дадени оценки от двата изпита.

Обучителните данни са във файла grades.txt, а Python кодът на скрипта е в lab6.py.

1. Визуализиране на обучителните данни

Задача 1. Визуализирайте обучителните данни в следната графика (фиг. 1):



Фиг. 1. Визуализиране на обучителните данни

2. Сигмоидна функция

В логаритмичната регресия хипотезата има следният вид:

$$h_\theta(x) = g(\theta^T x)$$

, където g се нарича сигмоидна функция със следната дефиниция:

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

, където:

$z = \theta^T x$ – параметър на сигмоидната функция

e – неперово число ($e \cong 2{,}72$).

За всяка стойност на параметъра z, $0 \leq g(z) \leq 1$.

Задача 2. Довършете функцията sigmoid(z), която изчислява стойност на сигмоидна функция с параметър z.

3. Функция на цената (cost function)

Функцията на цената в логаритмичната регресия с изчислява по следния начин:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log(h_\theta\left(x^{(i)}\right) + \left(1 - y^{(i)}\right)\log(1 - h_\theta\left(x^{(i)}\right))]$$

, където

$J(\theta)$ – функцията на цената с параметри вектор θ

m – брой входни данни

$x^{(i)}$ – входните данни (за оценки от два изпита)

$y^{(i)}$ – изходни стойности (1 – приет, 0 – не е приет)

log – функция логаритъм (с основа числото e)

Задача 3. Довършете функцията costFunction(theta, X, y), която изчислява цената за theta, X и y.

Резултат:

Cost:

0.69314718056

4. Градиент на цената

Градиентът на цената представлява вектор със същата размерност като вектора θ, като стойността на елемент j (за $j=0,1,2,\ldots,n$) се изчислява по следния начин:

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}\right) - y^{(i)})x_j^{(i)}$$

Обърнете внимание, че градиентът се изчислява по същия начин както и при линейната регресия. Разликата е, че хипотезата h_θ е различна, защото линейната и логаритмичната регресия имат различни дефиниции за h_θ(x).

Задача 4. Довършете функцията gradient(theta, X, y) за изчисляване на цената на градиента.

Резултат:

Grad:

[-0.1   -12.00921659 -11.26284221]

5. Намиране на минимум на функцията на цената

В предишните упражнения реализирахме градиентно спускане за намиране на стойности на параметрите θ, при които функцията на цената J има минимална стойност. В това упражнение ще използваме функцията *minimize* от библиотеката *scipy.optimize*. Извикваме функцията по следния начин:

res = minimize(costFunction, initial_theta, args=(X,y), method=None, jac=gradient, options={'maxiter':400})

Тази функция намира минимум на ф-та costFunction и съответните стойности на параметрите initial_theta.

Наблюдавайте резултата от изпълнението на тази функция в променливата *res*. Търсените стойности на параметрите са в *res.x*:

-25.1613, 0.206232, 0.201472

6. Оценяване на построената хипотеза

За да проверим построената хипотеза за логаритмична регресия, ще проверим каква е вероятността за приемане на студент с оценка 1 – 45 точки и оценка 2 – 87 точки.

Задача 5. Да се изчисли вероятността за приемане на студент с оценка 1 – 45 точки и оценка 2 – 87 точки.

Резултат:

0.776

Как се обяснява този резултат?

II. Логаритмична регресия с регуляризация

Регуляризацията е метод за предотвратяване на един от най-важните проблеми в машинното обучение - т.нар. припокриване (overfitting). Този проблем се изразява в точно (или почти точно) съвпадение на построената хипотеза с обучителните данни (т.е. точност близка до 100%). Това трябва да бъде предотвратено, защото ще доведе до големи отклонения при прилагане на хипотезата за нови (т.е. неизвестни при обучението) данни.

Разполагаме с данни за тестове на електронни чипове. В обучителните данни за всеки чип имаме резултат от изпитване 1, изпитване 2 и решение дали чипът е дефектен (1) или не е дефектен (0). Ще построим хипотеза, която предсказва дали чип ще бъде определен за годен на базата на резултатите от тест.

Обучителните данни са във файла chip_data.txt, а Python кодът на скрипта е в lab6_Regularization.py.

Регуляризацията се изразява в добавяне на нов терм, съдържащ параметър λ - параметър на регуляризацията. Целта е да се увеличи значението на параметрите θ_j, което съответно намалява значението на променливите x_j.

1. Регуляризирана функция на цената

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}[y^{(i)} \log\left(h_\theta\left(x^{(i)}\right) + \left(1 - y^{(i)}\right)\log\left(1 - h_\theta\left(x^{(i)}\right)\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

Векторна форма на регуляризираната функция на цената

$$J(\theta) = \frac{1}{m}\left((\log\left(g(X\theta)\right)^{\mathrm{T}}\right)y + (\log\left(1 - g(X\theta)\right)^{\mathrm{T}}(1-y)) + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

2. Регуляризиран градиент

$$\frac{\delta J(\theta)}{\delta \theta_j} = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(i)} + \frac{\lambda}{m}\theta_j$$

Векторна форма на регуляризирания градиент

$$\frac{\delta J(\theta)}{\delta \theta_j} = \frac{1}{m}X^T(g(X\theta) - y) + \frac{\lambda}{m}\theta_j$$

Регуляризация на параметър θ_0 не се прави.

Задача 6. Във функцията за изчисляване на цената добавете регуляризация.

Наблюдавайте графиките на построените хипотези при $\lambda = 0$, $\lambda = 1$ и $\lambda = 100$. В коя от хипотезите се наблюдава overfitting и при коя голямо отклонение (high bias)? Коя е най-адекватната хипотеза?