Stanford CS 229 - Machine Learning - Lecture 3

Giovani da Silva

June 2021

1 The Concept of Underfitting and Overfitting

Underfitting: Happens where there are patterns in the data that the algorithm is just failing to fit. The model cant find relations between the variables.

Overfitting:Happens when the algorithm is fitting the idiosyncrasies of this specific data set. The model "decorates" what he is going to do, and try to apply this.

Next: class of algorithms called non-parametric learning algorithms that will help to alleviate the need somewhat for you to choose features very carefully.

2 Parametric learning algorithm

Defined as an algorithm that has a fixed number of parameters that fit to the data. Its a non-parametric learning algorithm, so the number of parameters grows with m(size of the training set).

2.1 Locally weighted regression

The algorithm does linear regression in certain scopes of the function.

Will fit θ to minimize
$\sum_i w_i(y_i - \theta^T x_i)^2$
where $w_i = exp(-\frac{(x_i - x)^2}{2\tau^2})$

if $|x_i - x|$ is small, then w_i 1
if $|x_i - x|$ is large, then w_i 0

This algorithm is going to give the points close to x_i a large weight and give the points far away a small weight. So for the points that are far away,w_i will be close to zero. For the points that are far away, they will not contribute much to the summation.
The effect of using this weighting is that locally weighted linear regression fits a set of parameters theta, paying much more attention to fitting the points close by accurately.

The parameterτ, which is called bandwidth parameter, controls how fast the weights fall of with the distance.

Probabilistic Interpretation

Assume $y_i = \theta^T x_i + \epsilon_i$

ϵ_i = error, captures unmodeled effects(things we forgot to model), an random noise, etc.

Assuming error distributed by gaussian 0 to σ^2 or

$\epsilon_i \sim \mathcal{N}(0, \sigma^2)$

The density of ϵ_i , where σ is the standard deviation is:

$P(\epsilon_i) = \frac{exp(-\frac{\epsilon_i^2}{2\sigma^2})}{\sqrt{2\pi}\sigma}$

Assuming ϵ_i's are independent from each other

Where L is likelihood

$L(\theta) = P(\vec{y}(X, \theta) = \prod_{i=1}^{m} P(y_i|x_i, \theta) = \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(y_i - \theta^T x_i)^2}{2\sigma^2})$

Obs:The difference between likelihood and Probability is that we see likelihood as a function of θ

Maximum likelihood:

Choose θ (data) to maximize L(θ) as much as possible

being $l(\theta) = \log L(\theta)$

$\log \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(y_i - \theta^T x_i)^2}{2\sigma^2}) = \sum_{i=1}^{m} \log \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(y_i - \theta^T x_i)^2}{2\sigma^2})$

$= mlog\frac{1}{\sqrt{2\pi}\sigma} + \sum_{i=1}^{m} \log \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(y_i - \theta^T x_i)^2}{2\sigma^2})$

So, maximize l(σ) is the same of minimizing $\sum_{i=1}^{m} \frac{(y_i - \theta^T x_i)^2}{2}) = J(\theta)$

So, the algorithm of the previous lecture is also maximize likelihood, assuming there are errors in our data.

3 Classification algorithms

Generally, is bad to apply linear regression algorithms to classification problems, if you have one value that is far away from the others, you will have a bad prediction(think about a line in the function)

3.1

Where y $\in \{0, 1\}$
Yes or no, true or false,etc answers.
First thing we want to do is limit the hypothesis.
$h_\theta(x) \in [0, 1]$

Choose: $h_\theta(x) g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$
$g(z) = \frac{1}{1+e^{-z}}$

Above we have the Signoid function or Logistic function

Obs:one of the reasons we choose the Logistic function is the facility it has to generalize linear models.

Lets see a probabilistic interpretation for the hypothesis:

The hypothesis is outputting all these numbers that lie between zero and one

Think of hypothesis as trying to estimate the probability that Y is equal to one. And because Y has to be either zero or one then the probability of Y equals zero is going to be that.

So, $P(y = 1|x : \theta) = h_\theta(x)$
$P(y = 0|x : \theta) = 1 - h_\theta(x)$
$P(y = 1|x : \theta) = h_\theta(x)^y(1 - h_\theta(x))^{1-y}$
$L(\theta) = P(\vec{y}|X : \theta) = \prod_i h(x_i)_i^y(1 - h(x_i))^{1-y_i}$

As usual, we want to maximize L(θ) using l = log($l(\theta)$)

We can use what we did before, were
$\theta := \theta + \alpha\nabla_\theta l(\theta)$
To maximize the function.After some derivations, we have:
$\theta_j := \theta_j + \alpha \sum_{i=1}^m (y^i - h_\theta(x^i))x_j^i$

3.2 Digression Perception

Uses if you want to force G of Z to up the value to either zero or one.
g(z) = 1 if z >= 0

$g(z) = 0$ otherwise

This is very different flavor of algorithm than least squares regression and logistic regression, and, in particular, because it outputs only values are either zero or one it turns out it's very difficult to endow this algorithm with probabilistic semantics.