Gradient Descent

28 Feb 2023

Sumit Kumar Yadav

Department of Management Studies
Indian Institute of Technology, Roorkee

Recap and Today

- Logistic Regression - Even more discussion
- Softmax - More discussion
- Implementation of Logistic Regression/Softmax in Python
- Gradient Descent
- Doubts/Mid-sem Discussion

Logistic Regression - Main Ideas

Let
$w = [\alpha, \beta_1, \beta_2, ..., \beta_k]$
$x = [1, x_1, x_2, x_3, ..., x_k]$
$y = 1$, if Yes, $y = -1$ if No

$$\Pr[+1 \text{ given } x] = \frac{1}{1 + \exp(-w \cdot x)}$$

$$\Pr[-1 \text{ given } x] = \frac{1}{1 + \exp(+w \cdot x)}$$

Or more compactly

$$\Pr[y \text{ given } x] = \frac{1}{1 + \exp(-y \times w \cdot x)}$$

Maximum Likelihood Principle

Probability of observing the dataset as per the model is

$$\prod_i \Pr\left[y_i \text{ given } x_i\right] = \prod_i \frac{1}{1+\exp\left(-y_i w \cdot x_i\right)}$$

We are looking for w which will maximize this, or alternatively w which will minimize the expression below -

$$\min_w \sum_i \log\left(1+\exp\left(-y_i w \cdot x_i\right)\right)$$

We are looking for w which will maximize this, or alternatively w which will minimize the expression below -

$$\min_{w} \sum_{i} \log\left(1 + \exp\left(-y_i w \cdot x_i\right)\right)$$

How to find w??
Consider a simpler one-dimensional setup.(Remember the onion price example)
$w = [\alpha, \beta]$
Find the derivative of the expression w.r.t. α and β?

Simpler Case

We are looking for α and β which will minimize the expression below -

$$\min_{\alpha,\beta} \sum_i \log\left(1 + \exp\left(-y_i(\alpha + \beta x_i)\right)\right)$$

Simpler Case

We are looking for α and β which will minimize the expression below -

$$\min_{\alpha,\beta} \sum_i \log\left(1 + \exp\left(-y_i(\alpha + \beta x_i)\right)\right)$$

Partial Derivative w.r.t. alpha will give -

$$\sum_i \left(\frac{\exp\left(-y_i(\alpha + \beta x_i)\right)}{1 + \exp\left(-y_i(\alpha + \beta x_i)\right)} \right) y_i = 0$$

Simpler Case

We are looking for α and β which will minimize the expression below -

$$\min_{\alpha,\beta} \sum_i \log\left(1 + \exp\left(-y_i(\alpha + \beta x_i)\right)\right)$$

Partial Derivative w.r.t. alpha will give -

$$\sum_i \left(\frac{\exp\left(-y_i(\alpha + \beta x_i)\right)}{1 + \exp\left(-y_i(\alpha + \beta x_i)\right)}\right) y_i = 0$$

Partial Derivative w.r.t. beta will give -

$$\sum_i \left(\frac{\exp\left(-y_i(\alpha + \beta x_i)\right)}{1 + \exp\left(-y_i(\alpha + \beta x_i)\right)}\right) x_i y_i = 0$$

Simpler Case

We are looking for α and β which will minimize the expression below -

$$\min_{\alpha,\beta} \sum_i \log\left(1 + \exp\left(-y_i(\alpha + \beta x_i)\right)\right)$$

Partial Derivative w.r.t. alpha will give -

$$\sum_i \left(\frac{\exp\left(-y_i(\alpha + \beta x_i)\right)}{1 + \exp\left(-y_i(\alpha + \beta x_i)\right)}\right) y_i = 0$$

Partial Derivative w.r.t. beta will give -

$$\sum_i \left(\frac{\exp\left(-y_i(\alpha + \beta x_i)\right)}{1 + \exp\left(-y_i(\alpha + \beta x_i)\right)}\right) x_i y_i = 0$$

Are these two equations easy to solve analytically? (Like we were able to do in case of linear regression)

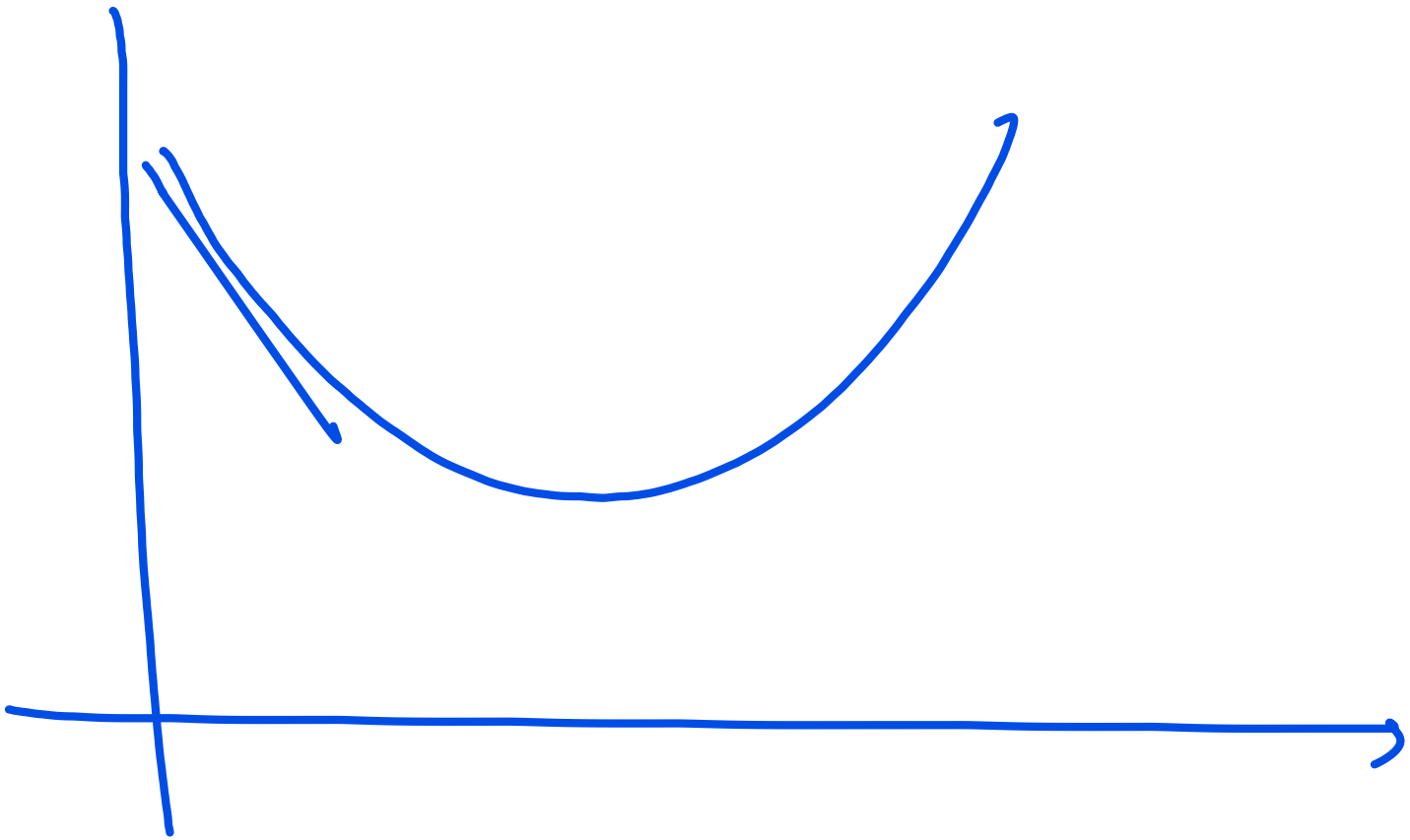
Gradient Descent

How can we find the minimum of the following function - f(x) = (x-2)(x-3)

Gradient Descent

How can we find the minimum of the following function - f(x) = (x-2)(x-3)
How about g(x) = (x-2)(x-3)(x-4)(x-5)(x-6)(x-7)

Convex Functions



Gradient Descent

Multidimensional Case

Frame Title

Frame Title

Implications of Linearity of Gradient Descent

Gradient is a linear operator, i.e. $\nabla(f_1 + f_2) = \nabla f_1 + \nabla f_2$ has great practical importance in machine learning.
The gradient of the entire function can be found by taking the sum of the gradient of the function on individual data points.

Stochastic Gradient Descent

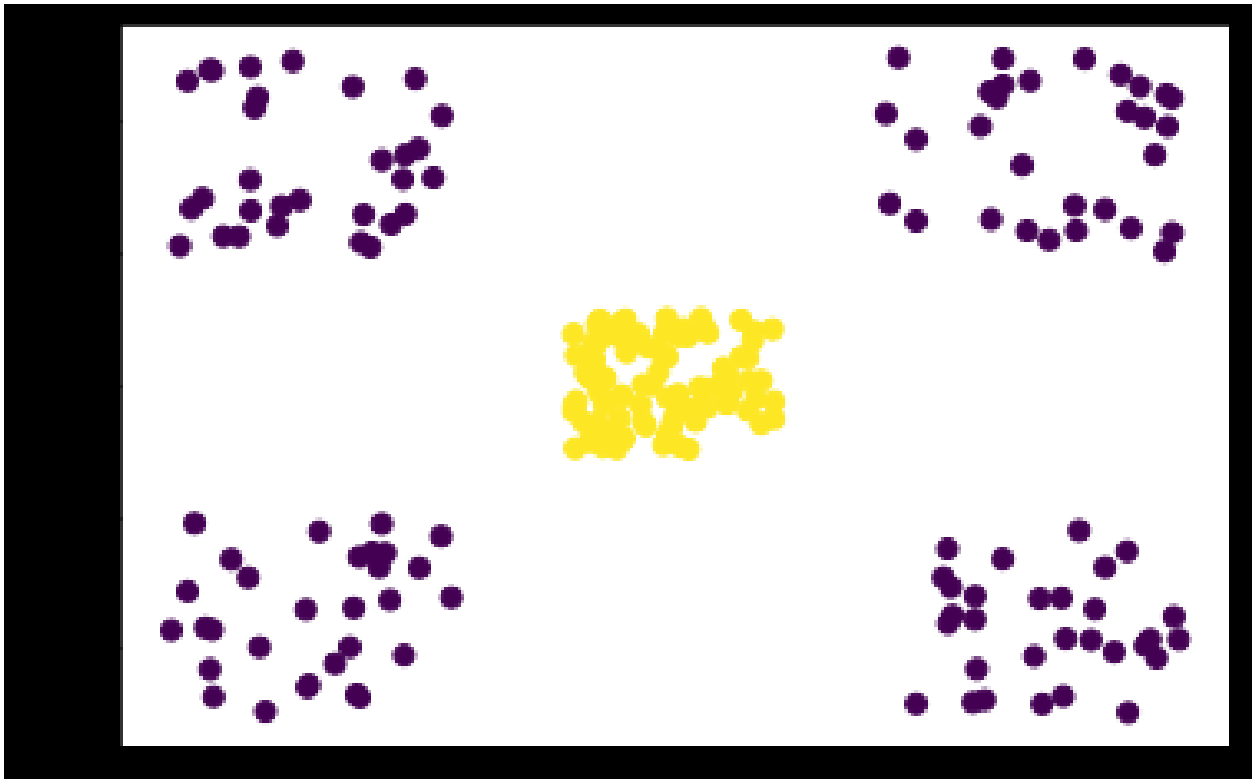
Because the goal is to minimize the function as a sum(or equivalently average) of the data points, if we have a million data points, in every iteration of the gradient descent, we would need to compute the gradient of one million data points and then add them up.
As an approximation, we can estimate the gradient instead of computing it for all data points. (Sampling)
The technique works because all data points are assumed to be sampled from the same distribution.

Federated Learning

1. Hospitals who wish to train an ML model on their pooled data, but who are forbidden by privacy laws to hand the data to other organizations.
2. Owners of Internet of Things (IoT) devices, who wish to benefit from training on their data but do not wish to submit the data.

Equivalent Cutoff condition with threshold

Let us say we keep the threshold as 0.5. What is the decision surface?



Decision Surface is a line (when data is 2-D, else a plane (or hyperplane))

For any threshold, logistic regression cannot be a good fit to this dataset because??

Multi-class classification - Model

multiple classes in the data

$y \in \{1, 2, 3, \cdots, r\}$

Instead of a single weight vector w, we consider r weight vectors $w_1, w_2, w_3, \cdots, w_r$.

$$\Pr[\text{ output } i \text{ given } x] = \frac{\exp(w_j \cdot x)}{\sum_{j=1}^{r} \exp(w_j \cdot x)}$$

Multi-class classification - Model

multiple classes in the data

$y \in \{1, 2, 3, \cdots, r\}$

Instead of a single weight vector w, we consider r weight vectors $w_1, w_2, w_3, \cdots, w_r$.

$$\Pr[\text{ output } i \text{ given } x] = \frac{\exp(w_j \cdot x)}{\sum_{j=1}^{r} \exp(w_j \cdot x)}$$

This is called the Soft-Max function, it converts a given set of numbers to probabilities.

Multi-class classification

S.No	X1	X2	Class
1	0.12	0.11	Green
2	0.14	0.44	Blue
3	0.47	0.50	Red
4	0.43	0.14	Green
5	0.37	0.40	Blue
6	0.13	0.49	Blue
7	0.41	0.13	Blue
8	0.16	0.12	Red
9	0.31	0.38	Green
10	0.22	0.29	Red

Consider a model with

$\alpha_g = 1, \beta_{g1} = 2, \beta_{g2} = 3$

$\alpha_b = 4, \beta_{b1} = -5, \beta_{b2} = 6$

$\alpha_r = 7, \beta_{r1} = 8, \beta_{r2} = -9$

What is the likelihood?

Multi-class classification

Let p_{ij} be the probability that the i^{th} data point is of type j
Let's consider the first data point for which X1 = 0.12, X2 = 0.11
As per the model, the probability of it being green can be computed as -

$$p_{1g} = \frac{e^{1+2*0.12+3*0.11}}{e^{1+2*0.12+3*0.11} + e^{4+(-5)*0.12+6*0.11} + e^{7+8*0.12+(-9)*0.11}}$$

Similarly,

$$p_{1b} = \frac{e^{4+(-5)*0.12+6*0.11}}{e^{1+2*0.12+3*0.11} + e^{4+(-5)*0.12+6*0.11} + e^{7+8*0.12+(-9)*0.11}}$$

And,

$$p_{1r} = \frac{e^{7+8*0.12+(-9)*0.11}}{e^{1+2*0.12+3*0.11} + e^{4+(-5)*0.12+6*0.11} + e^{7+8*0.12+(-9)*0.11}}$$

Example - Contd.

A similar exercise can be performed on all the data points. We get the following result.

S.No	Green	Blue	Red	Observed Class
1	0.004265	0.051441	0.944294	Green
2	0.029431	0.830513	0.140056	Blue
3	0.047325	0.158705	0.79397	Red
4	0.001005	0.001515	0.99748	Green
5	0.027343	0.136793	0.835864	Blue
6	0.025891	0.910426	0.063683	Blue
7	0.001005	0.001691	0.997304	Blue
8	0.003846	0.036124	0.96003	Red
9	0.028342	0.203231	0.768427	Green
10	0.0173	0.177809	0.804891	Red

Thus, the likelihood of observing the data given the model is -

$p_{1g} \cdot p_{2b} \cdot p_{3r} \cdot p_{4g} \cdot p_{5b} \cdot p_{6b} \cdot p_{7b} \cdot p_{8r} \cdot p_{9g} \cdot p_{10r}$

Softmax for 2 classes

Gradient Descent

Thank you for your attention