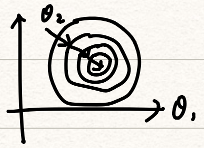
1. What is **gradient descent**? Why is it important?

GD : an iterative optimization algorithm (step-by-step)

→ find the params corresponding to optimal points of a target function (e.g. min loss, max likelihood, ...) step-by-step.

→ Start with initial param values, incrementally modify these values in the way that leads to largest improvement. (take derivatives!)



Important: for optimization problem with no closed from solution.

2. What is **Logistic Regression**? What is "logistic"? What are we "regressing"?

LR: $y=1$: positive; $y=0$: negative

Goal: train classifier that can make binary decision about the class of an input obs.

⇒ Given test instance $x: [x_1, \dots, x_f] \Rightarrow 1/0$.

⇒ Model: calculate prob $P(y=1|x)$

Decision boundary as 0.5 :

$$\text{classify as } \begin{cases} 1 & \text{if } P(y=1|x) > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

We apply logistic function (sigmoid) σ to regression z.

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

$$z = \theta_0 + \theta_1 x_1 + \dots + \theta_f x_f \quad (\theta_i : \text{params})$$

- Easy to calculate derivative ⇒ for gradient descent
- Has range $[0, 1]$ ⇒ estimate probability.

Regressing : log odds : $\log \frac{p}{1-p} = z$

3. Bob tries to gather information about this year's apple harvest and ran a search in his favorite online news outlet. He retrieved a number of articles but found that a large portion of the retrieved articles are about the Apple laptops and computers -- and hence irrelevant to his search. He wants to build a logistic regression classifier, which uses the counts of selected words in the news articles to predict the class of the news article (fruit vs. computer). He built the following data set of 5 training instances and 1 test instance. Develop a logistic regression classifier to predict label $\hat{y} = 1$ (fruit) and $\hat{y} = 0$ (computer).

ID	*apple*	*ibm*	*lemon*	*sun*	CLASS
	TRAINING INSTANCES				
A	1	0	1	5	1 FRUIT
B	1	0	1	2	1 FRUIT
C	2	0	0	1	1 FRUIT
D	2	2	0	0	0 COMPUTER
E	1	2	1	7	0 COMPUTER
	TEST INSTANCES				
T	1	2	1	5	?

For the moment, we assume that we already have an estimate of the model parameters, i.e., the weights of the 4 features (and the bias θ_0) is $\hat{\theta} = [\theta_0, \theta_1, \theta_2, \theta_3, \theta_4] = [0.2, 0.3, -2.2, 3.3, -0.2]$.

(i). Explain the intuition behind the model parameters, and their meaning in relation to the features

Feature engineering:

- Choose terms (as attributes)
- Define word occurence counts as attribute values.

LR:

$$P(y=1 \mid \underset{\sim}{x}) = \frac{1}{1+e^{-z}} = \sigma(z) \quad , \; z = \theta_0 + \theta_1 x_1 + \dots + \theta_f x_f$$

Params:

$\theta_1, \theta_2, \theta_3, \theta_4 \rightarrow$ importance of 4 features (terms) for predicting class 1 (fruit).

$\theta_0 \rightarrow$ bias (intercept)

(ii). Predict the test label.

$\hat{z} = \hat{\theta}_0 + \hat{\theta}_1 x_1 + \cdots + \hat{\theta}_4 x_4$

$= 0.2 + 0.3 - 2.2 \times 2 + 3.3 - 0.2 \times 5$

$= -1.6$

$\sigma(-1.6) = \frac{1}{1+e^{1.6}} = 0.17$ (for fruit)

$0.17 < 0.5 \Rightarrow$ Classify as computer

(iii). Recall the conditional likelihood objective

$$-\log \mathcal{L}(\theta) = -\sum_{i=1}^{n} y_i \log(\sigma(x_i;\theta)) + (1-y_i)\log(1-\sigma(x_i;\theta))$$

We want to make sure that the Loss (the negative log likelihood) our model, is lower when its prediction the correct label for test instance T, than when it's predicting a wrong label.

Compute the negative log-likelihood of the test instance (1) assuming that the true label y = 1 (fruit), i.e., our classifier made a mistake; and (2) assuming the true label as y = 0 (computer), i.e., our classifier predicted correctly.

Predict $\hat{y} = 0$ (computer) : (from (ii))

(1) If $y = 1$:

$-\log \mathcal{L}(\theta) = -\{1 \cdot \log(\sigma(x_i;\theta)) + 0 \cdot \log(1-\sigma(x_i;\theta))\}$

$= -\log(\sigma(x_i;\theta))$

$= -\log(0.17)$

$= 1.77$

(2) If $y = 0$:

$-\log \mathcal{L}(\theta) = -\{0 \cdot \log(\sigma(x_i;\theta)) + 1 \cdot \log(1-\sigma(x_i;\theta))\}$

$= -\log(1-\sigma(x_i;\theta))$

$= -\log(1-0.17)$

$= 0.19$ (lower loss)

4. For the model created in question 4, compute a single gradient descent update for parameter θ_1 given the training instances given above. Recall that for each feature j, we compute its weight update as

$$\theta_j \leftarrow \underbrace{\theta_j - \eta \sum_i (\sigma(x_i;\theta) - y_i)x_{ij}}_{-\frac{\partial}{\partial\theta}L(\theta)}$$

Summing over all training instances i. We will compute the update for θ_j assuming the current parameters as specified above, and a learning rate $\eta = 0.1$.

$\hat{\underset{\sim}{\theta}} = [0.2, 0.3, -2.2, 3.3, -0.2]$

① Compute $\sigma(x_i;\theta)$ for all i (training instances) (pred)

$\leftarrow x_{1A}$ to x_{1E}

$$\sigma(x_A;\theta) = \sigma(0.2 + (0.3 \times 1 + (-2.2) \times 0 + 3.3 \times 1 + (-0.2) \times 5)) = \underline{0.94}$$

$$\sigma(x_B;\theta) = \sigma(0.2 + (0.3 \times 1 + (-2.2) \times 0 + 3.3 \times 1 + (-0.2) \times 2)) = 0.97$$

$$\sigma(x_C;\theta) = \sigma(0.2 + (0.3 \times 2 + (-2.2) \times 0 + 3.3 \times 0 + (-0.2) \times 1)) = 0.65$$

$$\sigma(x_D;\theta) = \sigma(0.2 + (0.3 \times 2 + (-2.2) \times 2 + 3.3 \times 0 + (-0.2) \times 0)) = 0.03$$

$$\sigma(x_E;\theta) = \sigma(0.2 + (0.3 \times 1 + (-2.2) \times 2 + 3.3 \times 1 + (-0.2) \times 7)) = 0.12$$

② Update params (e.g. θ_1)

$$\theta_1 = \theta_1 - \eta \sum_{i \in \{A,B,C,D,E\}} (\sigma(x_i;\theta) - y_i)\, x_{1i}$$

$$\theta_1 = 0.3 - 0.1 \sum_{i \in \{A,B,C,D,E\}} (\sigma(x_i;\theta) - y_i)\, x_{1i}$$

η (learning rate)

$$\theta_1 = 0.3 - 0.1 \underbrace{[((\sigma(x_A;\theta) - y_A).x_{1A}) + ((\sigma(x_B;\theta) - y_B).x_{1B}) + ((\sigma(x_C;\theta) - y_C).x_{1C}) + ((\sigma(x_D;\theta) - y_D).x_{1D}) + ((\sigma(x_E;\theta) - y_E).x_{1E})]}_{\Sigma}$$

$$= 0.3 - 0.1 [((\underline{0.94} - 1) \times 1) + ((0.97 - 1) \times 1) + ((0.65 - 1) \times 2) + ((0.03 - 0) \times 2) + ((0.12 - 0) \times 1)]$$

$$= 0.3 - 0.1((-0.06) + (-0.03) + (-0.70) + 0.06 + 0.12) = 0.3 - 0.1\,(-0.61)$$
$$= 0.3 + 0.061 = 3.061 \quad \text{(new } \theta_1\text{)}$$

⇒ Do same thing for other θ's.

✱ Note: update all params at once.

5. [OPTIONAL] What is the relation between "odds" and "probability"?

prob : $p = p(\text{success})$

odds : ratio of p(success) to p(failure) : $\frac{p}{1-p}$

E.g. 8 balls : 5 red $\Rightarrow$ $p(\text{red}) = \frac{5}{8}$

odds of drawing red ball :

$$\text{odds} = \frac{\frac{5}{8}}{1-\frac{5}{8}} = \frac{\frac{5}{8}}{\frac{3}{8}} = \frac{5}{3} = 1.7$$

6. [OPTIONAL] (a) What is **Regression**? How is it similar to **Classification**, and how is it different?

(b) Come up with one typical classification task, and one typical regression task. Specify the range of valid values of *y* (results) and possible valid values for *x* (attributes).

(a) Both supervised learning methods, use labelled training dataset to make predictions.

Nominal target → Classification

Numeric target → Regression

(b) Regression : house price prediction

attributes : location, size, age, ...

class (y) : real value price (positive)

Classification : Sentiment analysis of movie reviews

attributes : set of words, author ID, length of review, ...

class (y) : Weak (1), Not bad (2), Good (3), Great (4), Master Piece (5)