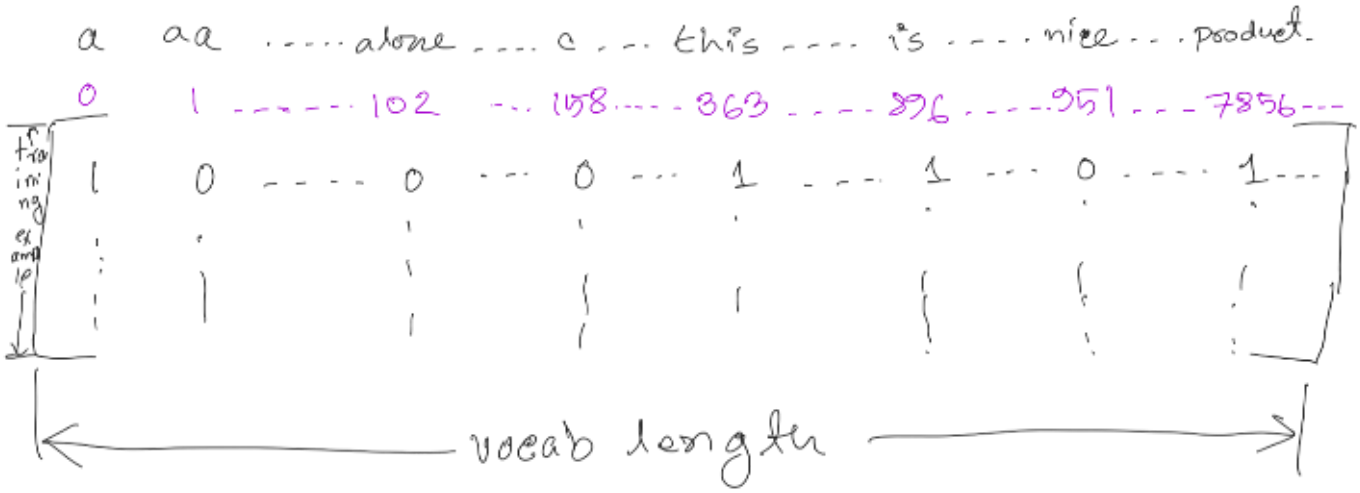
Naive Bayes

In GDA input values are continious but in Naive-Bayes input values are discrete. Lets say we want to build a review classifier where our model will predict if a given review is positive(1) or negetive(0). First we will convert our input into vector. For that we need to make a vocabulary which is nothing but a number representation of words.

This is a high quality product → positive

[363 896 0 3391 1001 7856] [1]

In our vocabulary word "This" is represented by number 363 We will convert this vector as one hot encoding. In one hot encoding a vector will have the same length as the length of vocabulary. If a particular word is present that position will have value 1. And rest of them will have value zero.



Having chosen our feature vector, we now want to build a discriminative model. So, we have to model p(x|y). But if we have, say, a vocabulary of 50000 words, then x ∈ {0, 1}^50000 (x is a 50000-dimensional vector of 0's and 1's), and if we were to model x explicitly with a multinomial distribution over the 250000 possible outcomes, then we'd end up with a

(250000−1)-dimensional parameter vector. This is clearly too many parameters. I

To reduce this huge parameter we will make a very strong assumption that is "x_i's are conditionally independent given y". If we wouldn't mention conditionally that would mean all the input feature are independent with each other. But the real assumption is input feature's are independent only for a particular output. Because of this assumption this algorithm is called Naive Bayes.

Bayes rule is:-

$$P(y|x) = \frac{P(x|y)\,P(y)}{P(x)}$$

To predict $P(y|x)$ we have to find each of the value :-

$P(x|y)$

$$P(x_1, \ldots x_{50,000} | y)$$

$$= P(x_1|y)\,P(x_2|y,x_1)\,P(x_3|y,x_1,x_2)\ldots P(x_{50,000}|y, x_1, \ldots x_{49999})$$

$$= P(x_1|y)\,P(x_2|y)\ldots P(x_{50,000}|y)$$

Because of the naive assumption.

$$= \prod_{i=1}^{n} P(x_i|y)$$

→ We need to calculate this for each of the output label.

For positive review and word "product":-

$P($ product $|y=1)$ = # occurance of "product" in all positive

$\phi_{product|y=1} = P(\ldots|y=1) = \frac{\ldots \text{review}}{\# \text{ number of positive review}}$

<u>Generally,</u>

$$\phi_{j|y=k} = \frac{\sum_{i=1}^{m} 1\{x_j^{(i)} = 1 \text{ and } y = k\}}{\sum_{i=1}^{m} 1\{y^{(i)} = k\}}$$

m = number of training example.

P(y)

this is the class probability. That means probability of a class occurance without seeing the input feature. This is also called the prior probability.

Prior probability for class k is.

$$P(y=k) = \frac{\text{Numer of occurance of class 'k'}}{\text{total training example}}$$

$$= \frac{\sum_{i=1}^{m} 1\{y^{(i)} = k\}}{m}$$

P(x)

$$P(x) = \sum_{i=1}^{k} \prod_{j=1}^{n} P(x_i | y=k)$$

Making prediction:-

We will calculate the prediction for each of the output class. for class 'k'

$$P(y=k | x) = \frac{P(x|y=k)\, P(y=k)}{P(x)}$$

[illegible]

whichever class which have the higest probability will consider that as output label.

We can apply naive bayes for continious value input. by **discretize** it.

Living area	< 400	400-800	800-1200	1200-1600	1600
x_i	1	2	3	4	5

Laplace smoothing :-

Let say we want to classify the review

"This product is waste of money".

Our training example didn't have any word "waste". So,

$P(x_{waste} | y = k) = 0$ for all value of 'k'

$$P(x | y=k) = P(x_0 | y=k) P(x_1 | y=k) \dots \underbrace{P(x_{waste} | y=k)}_{\downarrow 0} \dots$$

$$= 0$$

$$\therefore P(y|x) = \frac{\overbrace{P(x|y)}^{\to 0} P(y)}{\underbrace{P(x)}_{\to 0}}$$

$$= \frac{0}{0}$$

$$= \text{undefined}.$$

Laplace smoothing is used to solve this problem.

m

$$P(y=k) = \frac{\sum_{i=1}^{m} 1\{y^{(i)}=k\} + 1}{m + n}$$ — n number of output class.

$$P(x_j \mid y=k) = \frac{\sum_{i=1}^{m} 1\{y^{(i)}=k \text{ and } x_j^{(i)}=1\} + 1}{\sum_{i=1}^{m} 1\{y^{(i)}=k\} + n.}$$

Even if training example doesn't see any word it won't set it's probability to zero. Instead it will make that probability very small number.