__/__/__

Ritisha-Gupta, MT22056 | Assignment-4

Ques 3-(a) If one of the classes has zero training samples, then it will assign it 'zero' probabilities and frequency based probability estimate will be zero & this will get a zero when all the probabilities are multiplied. This is known as 'zero frequency problem'. It skews the whole performance of the classification.

An approach to overcome this problem is to ~~add~~ use Laplace Estimator. Laplace Smoothening adds a small positive number to all the counts.

Generally we add 'one' to the count for every attribute-value-class combination when an attribute value doesn't occur with every class value. This will lead to the removal of all the zero values from the classes and, at the same time, will not ~~at~~ impact the overall relative frequency of classes. This process of smoothing our data by adding a no is k/a additive smoothing or Laplace Smoothing.

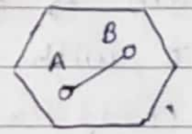
~~It'll be helpful if~~ It is a drawback/disadvantage of Naive Baye's Theorem. but it may be helpful to check whether a class has zero frequency or not. For that purpose we can use this.

__/__/____

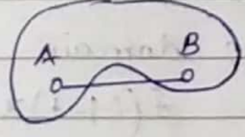
Ques 5a) • <u>Convex Set</u> → A set C is convex iff the line segment between any two points of C lies in C. That is $\forall u, v \in C$, $\forall\ 0 \le \theta \le 1$ we have

$$\boxed{\theta u + (1-\theta) v \in C}$$

Implies the line between any two points in the set C must also be fully contained within the set.



convex ✓



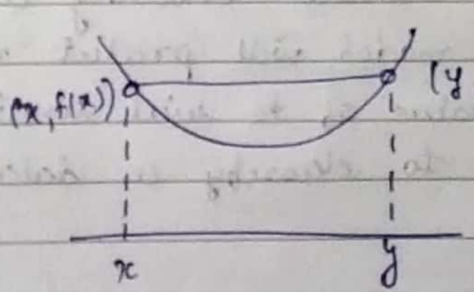
Non convex X

Properties of convex set -
- Intersection of convex sets is also convex.
- Projections onto convex sets are unique.

• <u>Convex Functions</u> → A convex function is a function defined on the convex domain such that, for any two points in the domain, the segment b/w the two points lies above the function curve between them.

A function $f: \mathbb{R}^n \to \mathbb{R}$ is convex if its domain is a convex set & $\forall x, \forall y$ in its domain, and all $\lambda \in [0,1]$, we have

$$\boxed{f(\lambda x + (1-\lambda) y) \le \lambda f(x) + (1-\lambda) f(y)}$$



→ convex Fun^c ✓

__/__/__

Lasso and Ridge Regularisation –

Ridge Regularisation is strictly convex and Lasso Regularisation is just convex.
First we need to understand the meaning of 'strictly convex' function. A function $f(x)$ is strictly convex on a convex domain if for every x_1 & x_2 in the domain & every t, with $0<t<1$,

$$f((1-t)x_1 + t x_2) < (t-1) f(x_1) + t f(x_2)$$

In lasso we are minimising $\|\beta_1\|$, which is not a strictly convex function.
Lasso is not strict because if x_1 and x_2 have the same sign, then the line segment & curve are exactly equal & ∴ they have ∞ many points in common.
In ridge regression, you're minimising $\|X\beta - y\|_2^2 + \lambda\|\beta\|_2^2$, which is a strictly convex function.

Ques 5b) Although the standard approach to choose 'k' is to try different values of k and whichever provides the best accuracy on our dataset is chosen. but here in-between $k=2$ & $k=3$ we'll choose $k=3$ because ~~KNN tries to approximate a locally smooth function~~. lower values of K in KNN, our model will predict more locally while choosing a larger value of K for KNN, our model will predict more globally.
Larger value of k will consider more number of neighbours to classify a datapoint to a clusture.

Ques 5c) Assuming training set T, consists of N points (x_i, y_i) which are Independent and identically, (IID)

Let the classifier be L_T which is trained on T.

so $y = L_T(n)$ is a distribution with mean 'μ' & variance 'σ^2' → Given

Total average loss over all n's $= E_{nT}[L(L_T(n), y)]$

Let squared loss, $S(h(n), y) = \frac{1}{2}(h(n) - y)^2$

$$E_{nT}[(L_T(n) - y)^2]$$

$$E_{nT}[(L_T(n) - E_T(L_T(n)) + E_T[L_T(n)] - y)^2]$$

$$E_{nT}[(L_T(n) - E_T(L_T(n)))^2 + (E_T(L_T(n)) - y)^2 + 2(L_T(n) - E_T(L_T(n))][E_T(L_T(n)) - y]]$$

$$\underbrace{E_{nT}[(L_T(n) - E_T(L_T(n)))^2]}_{\text{variance}} + \underbrace{E_x[(E + (L_T(n)) - y^2]}_{\text{bias}}$$

Let n models be trained on k subsets of D $\{D_i\}_{i=1}^{k}$

The bias term depends only on $E_T(L_T(n)]$.

$$L_T(n) = \frac{1}{k} \sum_{i=1}^{k} L_{T_i}(n)$$

$$E_T(L_T(n)) = \frac{1}{k} \sum_{i=1}^{n} E_{T_i}(L_{T_i}(n))$$

$$= \frac{1}{\cancel{k}} \cancel{k} \mu = \mu$$

Thus with ensembling bias term does not change. now,

$$L_T(n) = \frac{1}{k} \sum_{i=1}^{n} L_{T_i}(n) = var(L_T(m)) = var\left(\frac{1}{k} \sum_{i=1}^{k} L_{T_i}(n)\right)$$

$$[var(k_n) = v^2 var(n)] = \frac{1}{k^2} var\left(\sum_{i=1}^{k} L_{T_i}(n)\right)$$

(as m is IID) $\Rightarrow \frac{1}{k^2}\left(\sum_{i=1}^{k} var(L_{T_i}(n))\right)$

$$= \frac{1}{k^2} \times k \cdot L \cdot \sigma^2 = \frac{L}{k}\sigma^2$$

Scanned with CamScanner

_/_/_

now bias term does not change & we need to reduce the variance term to reduce loss.

$$\boxed{\frac{1}{k}\sigma^2} < \sigma^2$$

variance of ensemble model.

variance of single model to on entire data.

$$\Rightarrow \boxed{L < k}$$

∴ ~~The performance of the~~ ~~The no of model~~

∴ The no of learns for ensmble to perform better are more than L.