## 3. Logistic Regression

a) for N- class Instances.

→ for One-vs-All multi-class classification using logistic regression.

"N" binary classifier models are required.

→ for One-vs-One multi-class classification using logistic regression.

"$\frac{N(N-1)}{2}$" binary classifier model required.

## 4 Generalized Linear Model. (GLM)

(a) The only difference between Gamma Distribution & exponential distribution is that, the gamma distribution predicts wait time until the "k-th" event occur, while the exponential distribution predicts the wait time until very first event.

Derivation,

The CDF of a function is

$$P(X \leq t) = 1 - P(X > t)$$

$$= 1 - \sum_{i=0}^{k-1} \frac{(\lambda t)^i e^{-\lambda t}}{i!}$$

where $\lambda t \rightarrow$ poison rate

The PDF is $\frac{d}{dt}(CDF) = \frac{d}{dt}\left(1 - \sum_{i=0}^{k-1} \frac{(\lambda t)^i e^{-\lambda t}}{i!}\right)$

$$= \frac{d}{dt}\left(1 - e^{-\lambda t} - \sum_{i=1}^{k-1} \frac{(\lambda t)^i e^{-\lambda t}}{i!}\right)$$

$$= \lambda e^{-\lambda t} - \frac{d}{dt}\left(\sum_{i=1}^{k-1} \frac{(\lambda t)^i e^{-\lambda t}}{i!}\right)$$

$$= \lambda e^{-\lambda t} - \sum_{i=1}^{k-1} \frac{1}{i!}\left[i(\lambda t)^{i-1} \lambda . e^{-\lambda t} - \lambda(\lambda t)^i . e^{-\lambda t}\right]$$

$$= \lambda e^{-\lambda t} - \lambda e^{-\lambda t} \sum_{i=1}^{k-1} \frac{1}{i!}\left[i(\lambda t)^{i-1} - (\lambda t)^i\right]$$

$$= \lambda e^{-\lambda t} - \lambda e^{-\lambda t} \sum_{i=1}^{k-1} \left(\frac{(\lambda t)^i}{i!} - \frac{(\lambda t)^{i-1}}{(i-1)!}\right)$$

After expanding the summation, we get

$$\frac{d}{dt}(CDF) = \lambda e^{-\lambda t} + \lambda e^{-\lambda t}\left(\frac{(\lambda t)^{k-1}}{(k-1)!} - 1\right)$$

$$\boxed{\frac{d}{dt}(CDF) = \frac{\lambda . e^{-\lambda t} (\lambda t)^{k-1}}{(k-1)!}}$$

The final expression is same as pdf of exponential distribution, when $k = 1$.

The equation can also be written as

$$= \frac{\lambda . e^{-\lambda t} . (\lambda t)^{k-1}}{\Gamma(k)}$$

This shows that the gamma distribution follow poison process with a rate $\lambda$, & the wait time until $k$ arrivals follow $\Gamma(k, \lambda)$.

5

(a). Here nilesh is using covariance with KMeans & obtaining clusters using covariance.

i) KMeans algorithm clusters the points with respect to distance. If the distance of a point from let say cluster $C_1$ is more than cluster $C_2$, then KMeans places that point to cluster $C_2$.

But in this approach, there is a possibility that covariance indicates that point belong to $C_1$. So this approach may place point in wrong cluster.

ii) As KMeans uses distance & follow hard clustering & places the point to the cluster having minimum distance.

But covariance is kind of probabilistic to explain how points are related. So combinely, this combination of KMeans & covariance may not performs well!

7 —

a) Derivation of F5-score in terms of precision & recall

Fβ-score is an adjustable single-score metric used in machine learning for evaluating binary classification model using the precision & recall values for the +ve class.

The general formula for Fβ-score is :

$$F\beta\text{-score} = \frac{(1+\beta^2)(\text{precision} * \text{Recall})}{\beta^2 \text{precision} + \text{Recall}}$$

for F5-score, β = 5, we have

$$= \frac{(1+5^2)(\text{precision} * \text{Recall})}{5^2 \text{precision} + \text{Recall}}$$

$$\boxed{F5\text{-score} = \frac{26(\text{precision} * \text{Recall})}{25.\text{precision} + \text{Recall}}}$$

for value of α, we use Van Rijsbergen's effectiveness measure,

$$\alpha = \frac{1}{1+\beta^2} \Rightarrow \alpha = \frac{1}{1+5^2}$$

$$= \frac{1}{26} \Rightarrow \boxed{\alpha = 0.0384}$$

(b) Trade off between precision & recall.

Let's understand both precision & recall using the spam email example discussed in class.

| | Predicted No | Predicted Yes |
|---|---|---|
| Actual No | TN | FP |
| Actual Yes | FN | TP |

| | Spam | ham |
|---|---|---|
| Spam | 12 | 14 |
| ham | 0 | 114 |

precision tells out of ~~all~~ positive values, how many of them are actually correct.

The precision in this example is $\frac{TP}{TP+FP} = \underline{0.89}$

This means out of 100 spam emails 11 are marked ham.

→ Recall tells out of all actual positives, how many was identified correctly.

$$Recall = \frac{TP}{TP+FN} = \underline{1}$$

This means that the model correctly identifies 100% of the spam emails.

# Some observation → Improving precision typically reduces recall & vica-versa.

→ When the no. of false positive decreases, then false negative increases. As a result, recall decreases & precision increases.

→ When the no. of false positive increases, then false negative decreases. As a result, precision ~~increase~~ dec. & recall increases.

→ In F5-score, The value of 'α' is large, which means it gives less emphasis on precision & more on recall in the calculation of score.