MATHEMATICAL REPRESENTATION:

→ Principal Component Analysis:

Let's consider dataset $X = \{x_1, x_2 \ldots, x_n\}$, where $x_n \in R^D$ with 0 mean and covariance matrix given as,

$$S = \frac{1}{N} \sum_{n=1}^{N} (x_n - \bar{x})(x_n - \bar{x})^T$$

Where $\bar{x}$ is the mean row vector which is zero, Hence:

$$S = \frac{1}{N} \sum_{n=1}^{N} x_n * x_n^T$$

Covariance indicates the direction of linear relationship between variables and the matrix

S is the covariance matrix, λ is the eigen value and $\vec{v}$ is eigen vector. Therefore, principal components:

$$S\vec{v} = \lambda\vec{v}$$
$$S\vec{v} - \lambda\vec{v} = 0$$

$$\vec{v}(S - \lambda I) = 0$$

$(S - \lambda)I$ has to be non-invertible.

$$\det((S - \lambda I)) = 0$$

The eigen values are sorted in decreasing order to form the projection matrix

$$B = [\vec{v_1} \dots \vec{v_m}] \in R^{n \times m}$$

for $m \leq d$.

Projection matrix thus formed is

$$P = BB^T$$

New feature Subspace Z:

$$Z = BB^T X$$
$$Z = PX$$

→ Logistic Regression.

Loss Function:

$L(\hat{y}, y)$ = How much $\hat{y}$ differs from true value of y.

Let us consider,

$$\hat{y} = P(y = 1/x)$$

$\hat{y}_x$ is the probability that $y = 1$ given

$1-\hat{y} = P(y=0/x)$

$P(y/x) = \hat{y}^{y}(1-\hat{y})^{1-y}$

If $y=1 \Rightarrow P(y/x) = \hat{y}$

Taking the likelihood, $\log(\hat{y}^{y} * (1-\hat{y})^{(1-y)})$.

$y \log \hat{y} + (1-y)\log(1-\hat{y})$

$\Rightarrow -L(\hat{y}, y)$

$\Rightarrow \log P(y/x) = -L(\hat{y}, y)$.

The negative log likelihood is to maximize the probability by minimizing loss function.

MLE :

Since we now are using more than two classes, the log of the maximum likelihood function becomes:

$$L(\beta_i^T X) = \sum_{i=1}^{N} \log(p_i(x_i, \beta_i^T))$$

$$= \sum_{i=1}^{N} \log\left(\frac{e^{\beta_i^T x_i}}{1+e^{\beta_i^T x_i}}\right)$$

The Gradient:

The derivation of the gradient of the maximum likelihood function below:

$$\frac{\partial L(\beta, x_i)}{\partial \beta} = \sum_{i=1}^{N} \frac{\partial \cdot y_i \beta^T x_i - \partial \ln(1 + e^{\beta^T x_i})}{\partial \beta}$$

$$= \sum_{i=1}^{N} y_i x_i - \frac{x_i e^{\beta^T x_i}}{1 + e^{\beta^T x_i}}$$

$$= \sum_{i=1}^{N} x_i (y_i - p_i)$$

→ Classification

* Linear SVM:

SVM hypothesis:

$$h_{w,b}(x) = g(w^T x + b)$$

class labels are denoted as from 0 to 6,

functional margin and geometric margin,
optimization problem that the SVM solves

$$\phi(w) = \frac{1}{2} w^T w - \text{minimize}$$

Subject to $d_i (w^T x + b) \geq 1 \; \forall i$

equation for seperating hyperplane, w is the normal to the hyperplane

$$\pi : w^T x^{(i)} + b = 0$$

Functional margin of a hyperplane w.r.t ith training example is defined as:

$$\hat{\gamma}^{(i)} = y^{(i)} (w^T x^{(i)} + b)$$

geometric margin of a hyperplane w.r.t ith training example,

$$\gamma^{(i)} = y^{(i)} \frac{(w^T x^{(i)} + b)}{\|w\|}$$

This makes our optimization problem as

$$(w^*, b^*) = \text{argmax}_{w,b} \frac{2}{\|w\|}$$

Such that,

$$y^{(i)} (w^T x^{(i)} + b) \geq 1$$

→ Multinomial Naive Bayes

The dependent feature vector is $(x_1, x_2 \ldots x_n)$ and the class is C_k.

$$P(C_K | x_1 \dots x_n) = \frac{P(C_K)\, P(x_1 \dots x_n | C_K)}{P(x_1, \dots x_n)}$$

Now, for each given class C_K, each feature vector x_i is continuously independent of other feature.

$$P(x_i | C_K, x_i \dots x_n) = P(x_i | C_K)$$

This can be simplified as,

$$P(C_K | x_1 \dots x_n) = \frac{P(C_K) \prod_{i=1}^{n} P(x_i | C_K)}{P(x_1, \dots x_n)}$$

Since $P(x_1, \dots, x_n)$ is constant,

$$P(C_K | x_1, \dots, x_n) \propto P(C_K) \prod_{i=1}^{n} P(x_i | C_K)$$

$$\hat{y} = \underset{K}{\operatorname{argmax}}\, P(C_K) \prod_{i=1}^{n} P(x_i | C_K)$$

Log probabilities can be used to avoid underflows,

$$\hat{y} = \underset{K}{\operatorname{argmax}} \left(\ln P(C_K) + \sum_{i=1}^{n} \ln P(x_i | C_K) \right)$$

The parameter θ_K is estimated by maximum likelihood

$$\hat{\theta}_{Ki} = \frac{N_{Ki} + \alpha}{N_K + \alpha n}$$

$$\therefore \hat{y} = \arg\max_{k} \left(\ln P(C_k) + \sum_{i=1}^{n} \ln \frac{N_{ki} + \alpha}{N_k + \alpha n} \right)$$