Naive Bayes Classifier

→ Generative Counter-part to Logistic Regression

→ This also outputs the $P(y|x)$. (But this uses Bayes's Thm to do that).

→ Simple, yet powerful → used in document classification
→ spam filtering.

→ This makes use of the Strong (Naive) Conditional Independence assumption between the features of a given a label.

Suppose → $D = (X, \vec{y}) = \{(\vec{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$ (dimensions: X is (n, m), $\vec{y}$ is $(n, 1)$)

This enables us to express joint probabilities of individual features given a label.

$\vec{x} = (x_1, x_2, \ldots, x_m)$

($y^{(i)} \in \{0, 1\}$ for binary classification)

$$P(\vec{x}|y) = P(x_1, x_2, \ldots, x_m|y)$$

$$= P(x_1|y) \cdot P(x_2|y) \cdots P(x_m|y) \quad \text{—— (*)}$$

→ NB classifier predicts probability, $P(y|x)$, of a class label (y) given a feature vector (x) using Bayes' Theorem.

Class Conditional density

class prior

$$\underbrace{P(y|\vec{x})}_{\text{Posterior Probability}} = \frac{P(\vec{x}, y)}{P(\vec{x})} = \frac{P(\vec{x}|y) \cdot P(y)}{\underbrace{P(\vec{x})}_{\text{Evidence}}}$$

If the different possible labels are $\{y_1, y_2, \ldots, y_k\}$, then →

$$P(\vec{x}) = \sum_{r=1}^{k} P(\vec{x}, y_r) = \sum_{r=1}^{k} P(\vec{x}|y_r) \cdot P(y_r) \quad \text{—— (†)}$$

Now, using the naive Baye's Assumption (*) and (†) we have →

$$P(y=y_c|\vec{x}) = \frac{P(\vec{x}|y_c) \cdot P(y_c)}{P(\vec{x})}$$

$$= \frac{P(y_c) \cdot P(x_1|y_c) \cdot P(x_2|y_c) \cdots P(x_m|y_c)}{\sum_{r=1}^{k} P(\vec{x}|y_r) \cdot P(y_r)}$$

$$= \frac{P(y_c) \cdot P(x_1|y_c) \cdot P(x_2|y_c) \cdots P(x_m|y_c)}{\sum_{r=1}^{k} P(y_r) \cdot P(x_1|y_r) \cdot P(x_2|y_r) \cdots P(x_m|y_r)}$$

$$P(y=y_c|\vec{x}) = \frac{P(y_c) \cdot \prod_{j=1}^{m} P(x_j|y_c)}{\sum_{r=1}^{k} P(y_r) \prod_{j=1}^{m} P(x_j|y_r)}$$

The parameters of NB classifiers →

'k' Prior probabilities : $P(y_1), P(y_2) \cdots P(y_k)$

'k×m' Class conditional densities :

CD given $y=y_1$ ← $P(x_1|y_1), P(x_2|y_1) \cdots P(x_m|y_1)$

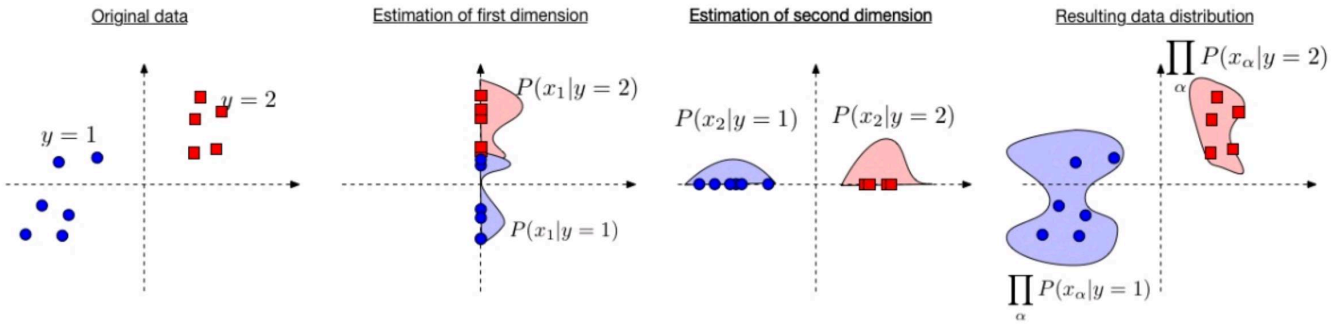
CD given $y=y_2$ ← $P(x_1|y_2), P(x_2|y_2) \cdots P(x_m|y_2)$

⋮

Conditional density given $y=y_k$ ← $P(x_1|y_k), P(x_2|y_k) \cdots P(x_m|y_k)$

* The number of parameters for each conditional density varies and depends on its mathematical form.

NB schematic



Modelling Conditional Densities. $P(x_j|y_c)$

This depends on nature of feature x_j

⇒ Binary Feature

→ $x_j \in \{0, 1\}$

$P(x_j|y_c)$ is Bernoulli (μ_{jc})

↓ Generalization

⇒ Categorical Feature

→ $x_j \in \{v_1, v_2, \dots v_e\}$

$P(x_j|y_c)$ is Cat $(e, \mu_{j1c}, \mu_{j2c}, \dots \mu_{jec})$

Multinomial Feature

→ Eg → Word count c_i as features with additional constraints that $\sum_{i=1}^{m} c_i = l$, the length of the sequence they represent.

$\vec{x} = (c_1, c_2, \dots, c_m)$ $\quad c_i \in \mathbb{N}$

$c_1 + c_2 + \dots + c_m = l.$

$P(\vec{x}|y_c)$ is Multinomial $(l, \mu_{1c}, \mu_{2c}, \dots \mu_{mc})$

Continuous Feature

→ features are real no.s.

$x_j \in \mathbb{R}$

$P(x_j|y_c)$ is $N(\mu_{jc}, \sigma_{jc})$

→ We need to estimate parameters of relevant distributions, one for each feature, for each class label.

→ We denote the set of parameters as a single vector $\vec{w}$ (of unknown dimension as this varies from problem to problem).

$\vec{w}$: Set of all parameters → Prior density + Class conditional density.

- **Bernoulli Distribution.**

$$P(x_j \mid y_c) \text{ is Bernoulli } (\mu_{jc})$$

$x_j \in \{0, 1\}$. Parameters: μ_{jc}

$$P(x_j = 1 \mid y_c) = \mu_{jc} \qquad P(x_j = 0 \mid y_c) = 1 - \mu_{jc}$$

$$P(x_j \mid y_c ; \mu_{jc}) = \mu_{jc}^{x_j} \cdot (1 - \mu_{jc})^{(1 - x_j)}$$

If there are 'k' classes:
- For each binary feature, we need 'k' parameters $\{\mu_{jc}\}_{c=1}^{k}$
- For 's' binary features, we need 'k x s' parameters

- **Categorical Distribution.**

Generalized Version of 'Bernoulli'.

Suppose $x_j \in \{V_1, V_2, \ldots, V_e\}$ (There are 'e' possible categories)

Parameters: $\vec{\mu_{jc}} = (\mu_{j1c}, \mu_{j2c}, \ldots, \mu_{jec})$

be the parameter vector for $P(x_j \mid y_c)$

Then →

$$P(x_j = V_1 \mid y_c ; e, \vec{\mu_{jc}}) = \mu_{j1c}$$

$$P(x_j = V_2 \mid y_c ; e, \vec{\mu_{jc}}) = \mu_{j2c}$$

$$P\left(x_j = v_e \mid y_c ; e, \vec{\mu_{jc}}\right) = \mu_{jec}$$

Then →

$$P\left(x_j \mid y_c ; e, \vec{\mu_{jc}}\right) = \mu_{j1c}^{1_{(x_j = v_1)}} \cdot \mu_{j2c}^{1_{(x_j = v_2)}} \cdots \mu_{jec}^{1_{(x_j = v_e)}}$$

This forces the constraint → $\sum_{\alpha=1}^{e} \mu_{j\alpha c} = 1$.

$$\left(\text{or } \text{sum}\left(\vec{\mu_{jc}}\right) = 1\right)$$

- If there are 'k' classes →
  - For each class 'c' we need $\text{len}\left(\vec{\mu_{jc}}\right)$ of parameters
  - For all 'k' classes → $k \times \text{len}\left(\vec{\mu_{jc}}\right)$
  - For all categorical features (if all 'm' are categorical)

$$k \cdot \sum_{j=1}^{m} \text{len}\left(\vec{\mu_{jc}}\right)$$

- **Multinomial Distribution**

Ex → When $\vec{x}$ is a count vector, i.e. each component 'x_j' is a "count of appearance" in the object it represents and sum $(\vec{x}) = \sum_{j=1}^{m} x_j = l$,

which is the length of the object.

Here, multinomial distribution is useful to model $P(\vec{x} \mid y_c)$. $\quad \vec{x} = (x_1, x_2, \ldots, x_m)$

Parameters: $l, \mu_{1c}, \mu_{2c}, \ldots, \mu_{mc}$
(for $P(\vec{x} \mid y_c)$)

constraint: $\sum_{i=1}^{m} x_i = l \qquad x_i \in \mathbb{N}.$

$$P\left(\vec{x} \mid y_c ; l, \vec{\mu_c}\right) = \frac{n!}{x_1! x_2! \cdots x_m!} \prod_{j=1}^{m} \mu_{jc}^{x_j}$$

where

$$\vec{\mu_c} = \left(\mu_{1c}, \mu_{2c}, \ldots, \mu_{mc}\right)$$

- If there are 'k' classes:
  - parameters per class (for class y_c): $m = \text{len}\left(\vec{\mu_c}\right)$
  - Total: $k \times m$

- **Gaussian Distribution.**

→ When x_j is a real no. (or continuous feature).

Parameters: μ_{jc}, σ_{jc}^2

(for $P(x_j|y_c)$)

$$P\left(x_j \mid y_c ; \mu_{jc}, \sigma_{jc}^2\right) = \frac{1}{\sqrt{2\pi}\,\sigma_{jc}} e^{-\frac{1}{2}\left(\frac{x_j - \mu_{jc}}{\sigma_{jc}}\right)^2}$$

- For each class, no. of parameters: $2 \times m$

$\vec{\mu}_c$, $\vec{\sigma}_{jc}$ → each of length m.

- for 'k' classes: $2 \times m \times k$.

↳ This is the one-dimensional form. We can also look at multi-variate form to gain direct access to $P(\vec{x}|y_c)$.

- **Multi-variate Gaussian:**

Parameters: $(\vec{\mu}_c)_{m\times 1}$, $(\Sigma_c)_{m\times m}$

$\vec{\mu}_c$: Mean vector Σ_c: Covariance Matrix.

In Naive Bayesian setting, where features are independent of each other given a label, we have the non-diagonal entries of Σ_c to be zero.

So, $(\Sigma_c)_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ (\sigma_c)_{ii}^2 & \text{if } i = j \end{cases}$

$$P(\vec{x} \mid y_c; \vec{\mu}_c, \Sigma_c) = \frac{1}{\sqrt{(2\pi)^m |\Sigma|}} e^{-\frac{1}{2}(\vec{x}-\vec{\mu})^T \Sigma^{-1} (\vec{x}-\mu)}$$

→ Assume 'k' classes:

- Parameters per class: $2\times m$.

$\vec{\mu}_c$, $\Sigma_c \rightarrow \begin{bmatrix} (\sigma_c)_{11} & & 0 \\ & (\sigma_c)_{22} & \\ 0 & & (\sigma_c)_{mm} \end{bmatrix}$

- Total: $2\times m \times k$.

- ## Loss function and Optimization.

→ Maximum Likelihood
→ Maximum log likelihood

) Estimates.

so, basically given a sample examples and labels →

$$\left(\vec{x}^{(i)}, y^{(i)}\right)^{n}$$

$$L(\vec{w}) = \prod_{i=1}^{n} P\left(\vec{x}^{(i)}, y^{(i)}; \vec{w}\right) \longrightarrow \text{Likelihood}$$

$$l(\vec{w}) = \sum_{i=1}^{n} \log\left(P\left(\vec{x}^{(i)}, y^{(i)}; \vec{w}\right)\right)$$

log Likelihood.

Our goal is to find the set of parameters aka $\vec{w}$ that maximizes $L(w)$ or $l(\vec{w})$.

Loss function

$$\ell(\vec{w}) = \sum_{i=1}^{n} \log\left(P(\vec{x}^{(i)}, y^{(i)}; \vec{w})\right)$$

We want to maximize the log likelihood.

Instead we can minimize negative log-likelihood →

$$J(\vec{w}) = -\ell(\vec{w})$$

$$= -\sum_{i=1}^{n} \log\left(P(x^{(i)}, y^{(i)}; \vec{w})\right)$$

Instead of doing that, we focus on maximizing the log likelihood.

$$\ell(\vec{w}) = \sum_{i=1}^{n} \log\left(P\left(\vec{x}^{(i)}, y^{(i)}; \vec{w}\right)\right)$$

$$= \sum_{i=1}^{n} \log\left(P\left(\vec{x}^{(i)} \mid y^{(i)}; \vec{w}\right) \cdot P\left(y^{(i)}; \vec{w}\right)\right)$$

Assuming → $\vec{x}^{(i)} = \left(x_1^{(i)}, x_2^{(i)}, \ldots, x_m^{(i)}\right)$

and using NB assumption.

$$= \sum_{i=1}^{n} \log\left(P\left(y^{(i)}; \vec{w}\right) \cdot \prod_{j=1}^{m} P\left(x_j^{(i)} \mid y^{(i)}; \vec{w}\right)\right)$$

$$= \sum_{i=1}^{n} \log\left(P\left(y^{(i)}; \vec{w}\right)\right) + \sum_{i=1}^{n}\sum_{j=1}^{m} \log\left(P\left(x_j^{(i)} \mid y^{(i)}; \vec{w}\right)\right)$$

$P\left(y^{(i)}; \vec{w}\right)$ → Prior probability → This comes from empirical probability from the sample.

$P\left(x_j^{(i)} \mid y^{(i)}; \vec{w}\right)$ → class-conditional density.

This depends on the probability distribution, depending on the type of features we are dealing with (as described above).

→ Optimization and Parameter estimation

Parameter estimation can be done in these 3 steps → (Maximum Likelihood Estimate)

(∓)

1. Calculate partial derivative of $l(\vec{w})$ w.r.t. each parameter (w_i).
2. Set partial derivative to '0', which is the condition at maxima.
3. Solve the resulting eq^n to obtain the parameter value $(\hat{w}_i)$.

---

① Estimating Prior Probabilities $P(y^{(i)}; \vec{w})$

let

$$w_{y^{(i)}} := P(y^{(i)} = y_c) = \frac{|\{i \in [n] \mid y^{(i)} = y_c\}|}{n}$$

$$= \frac{\sum_{i=1}^{n} 1_{(y^{(i)} = y_c)}}{n}$$

$$= \frac{\#\text{examples with labels } y_c}{n}$$

This is just the empirical probability.

② Estimating Class Conditional Densities.

- **Bernoulli Distribution.** (Pg no. - 6)

$$L(\vec{w}) = \sum_{i=1}^{n} \log\left(P(y^{(i)}; \vec{w})\right) + \sum_{i=1}^{n}\sum_{j=1}^{m} \log\left(P(x_j^{(i)} | y^{(i)}; \vec{w})\right)$$

Now, $x_j \in \{0,1\}$

$$P(x_j | y_c ; w_{jc}) = w_{jc}^{x_j} (1 - w_{jc})^{(1-x_j)}$$

$$\Rightarrow L(w) = \sum_{i=1}^{n} \log\left(w_{y^{(i)}}\right) + \sum_{i=1}^{n}\sum_{j=1}^{m} \log\left(w_{j y^{(i)}}^{x_j^{(i)}} \left(1 - w_{j y^{(i)}}\right)^{1 - x_j^{(i)}}\right)$$

$$= \sum_{i=1}^{n} \log\left(w_{y^{(i)}}\right) +$$

$$\sum_{i=1}^{n}\sum_{j=1}^{m} \left(x_j^{(i)} \cdot \log\left(w_{j y^{(i)}}\right) + \left(1 - x_j^{(i)}\right) \log\left(1 - w_{j y^{(i)}}\right) \right)$$

(Parameters for label y_r: $w_{1y_r}, w_{2y_r}, w_{3y_r}, \ldots, w_{my_r}$)

Now, we employ $(\mp) \rightarrow$ for $j=t$ and $y^{(i)}=y_r$

STEP 1:

$$\frac{\partial\left(\ell(\vec{w})\right)}{\partial w_{ty_r}} = \frac{\partial}{\partial w_{ty_r}}\left(\ell(\vec{w})\right) = \frac{\partial}{\partial w_{ty_r}}\left(\sum_{i=1}^{n} \log\left(w_{y^{(i)}}\right)\right) + \quad (\to 0)$$

↳ These are constants.

$$+\frac{\partial}{\partial w_{ty_r}} \sum_{i=1}^{n}\sum_{j=1}^{m}\left(x_j^{(i)} \cdot \log\left(w_{jy^{(i)}}\right) + \left(1-x_j^{(i)}\right)\log\left(1-w_{jy^{(i)}}\right)\right)$$

We only keep the relevant terms with $\left(y^{(i)}=y_r\right)$

$$= \sum_{i=1}^{n} 1_{\left(y^{(i)}=y_r\right)}\left(\sum_{j=1}^{m}\frac{\partial}{\partial w_{ty_r}}\left(x_j^r \cdot \log\left(w_{jy_r}\right) + \left(1-x_j^r\right)\log\left(1-w_{jy_r}\right)\right)\right)$$

$$= \sum_{i=1}^{n} 1_{\left(y^{(i)}=y_r\right)}\left(\frac{x_t^r}{w_{ty_r}} - \frac{1-x_t^r}{1-w_{ty_r}}\right)$$

Only terms with $j=t$ survive.

STEP 2: Setting $\frac{\partial}{\partial w_{ty_r}}\left(\ell(\vec{w})\right) = 0$

$$\Rightarrow \sum_{i=1}^{n} 1_{\left(y^{(i)}=y_r\right)} \left(\frac{x_t^r}{w_{ty_r}} - \frac{1-x_t^r}{1-w_{ty_r}} \right) = 0$$

$$\Rightarrow \frac{\sum_{i=1}^{n} 1_{\left(y^{(i)}=y_r\right)} \left(x_t^r \left(1-w_{ty_r}\right) - \left(1-x_t^r\right) w_{ty_r} \right)}{w_{ty_r} \cdot \left(1-w_{ty_r}\right)} = 0$$

$$= \sum_{i=1}^{n} 1_{\left(y^{(i)}=y_r\right)} \left(x_t^r \left(1-w_{ty_r}\right) - \left(1-x_t^r\right) w_{ty_r} \right) = 0$$

$$= \sum_{i=1}^{n} 1_{\left(y^{(i)}=y_r\right)} \left(x_t^r - w_{ty_r} \right) = 0$$

$$= \sum_{i=1}^{n} 1_{\left(y^{(i)}=y_r\right)} x_t^r = \sum_{i=1}^{n} 1_{\left(y^{(i)}=y_r\right)} w_{ty_r}$$

$$= \boxed{w_{ty_r} = \frac{\sum_{i=1}^{n} 1_{\left(y^{(i)}=y_r\right)} x_t^r}{\sum_{i=1}^{n} 1_{\left(y^{(i)}=y_r\right)}}}$$

For binary type features
i.e. if $x_t^{(i)} \in \{0,1\}$
we can simplify this even more →

$$w_{t y_r} = \frac{\sum_{i=1}^{n} 1_{(y^{(i)} = y_r)} \, x_t^r}{\sum_{i=1}^{n} 1_{(y^{(i)} = y_r)}}$$

$\sum_{i=1}^{n} 1_{(y^{(i)} = y_r)}$ → # of examples with $y^{(i)} = y_r$

$\sum_{i=1}^{n} 1_{(y^{(i)} = y_r)} \, x_t^r$ → # of examples with $x_t^{(i)} = 1$ and $y^{(i)} = y_r$

PROBLEM WITH ZERO COUNT:

→ Now it becomes problematic if $\sum_{i=1}^{n} 1_{(y^{(i)} = y_r)} \, x_t^r = 0$

(i.e. if there is no example with $x_t^{(i)} = 1$ and $y^{(i)} = y_r$ at the same time)

This makes $w_{t y_r} = 0 \Rightarrow P(x_t \mid y_r) = 0$

$\Rightarrow P(y = y_r \mid \vec{x}) = 0$

FIXING PROBLEM WITH ZERO COUNT:

→ when $c=1$.

We use something called "Laplace Smoothing".

We can control this by adding a '$+c$' to numerator and '$+2c$' to the denominator.

$$w_{ty_r} = \frac{\left(\sum_{i=1}^{n} 1_{(y^{(i)}=y_r)}\, x_t^r\right) + c}{n_r + 2c}$$

$$n_r = \left(\sum_{i=1}^{n} 1_{(y^{(i)}=y_r)}\right)$$

c: This is a hyperparameter that helps control overfitting.

Too high value of 'c' leads to underfitting.

- ## Categorical Distribution — Pg no. - 6.

Suppose $x_j \in \{V_1, V_2, \ldots, V_e\}$ → There are 'e' possible values feature 'x_j' can take.

Parameters: $\vec{\mu}_{jc} = (\mu_{j1c}, \mu_{j2c}, \ldots, \mu_{jec})$

be the parameter vector for $P(x_j | y_c)$

Then →

$$P(x_j = V_1 | y_c ; e, \vec{\mu}_{jc}) = \mu_{j1c}$$
$$P(x_j = V_2 | y_c ; e, \vec{\mu}_{jc}) = \mu_{j2c}$$
$$\vdots$$
$$P(x_j = V_e | y_c ; e, \vec{\mu}_{jc}) = \mu_{jec}$$

Then →

$$P(x_j | y_c ; e, \vec{\mu}_{jc}) = \mu_{j1c}^{1_{(x_j = V_1)}} \cdot \mu_{j2c}^{1_{(x_j = V_2)}} \cdots \mu_{jec}^{1_{(x_j = V_e)}}$$

This forces the constraint → $\sum_{\alpha=1}^{e} \mu_{j\alpha c} = 1$.

(or $\text{sum}(\vec{\mu}_{jc}) = 1$)

Here, for a fixed $y^{(i)} = y_r$:

(i.e for a fixed class →)

$$\vec{w}_r = (\vec{\mu}_{1r}, \vec{\mu}_{2r}, \ldots \vec{\mu}_{mr})$$

where →

$$\vec{\mu_{jr}} = \left(\mu_{j1r} , \mu_{j2r} , \cdots \mu_{jer} \right)$$

Total no. of parameters → $n \times m \times e$ (if each parameter has 'e' choices)

$$\mu_{jvy_r} = P\left(x_j = v \mid y_r \right)$$

$$= \frac{\left(\sum_{i=1}^{n} 1_{\left(y^{(i)} = y_r\right)} \, 1_{\left(x_j^{(i)} = v\right)} \right) + c}{n_r + ce}$$

↳ When 'c = 1' → Laplace Smoothing.

$$n_r = \left(\sum_{i=1}^{n} 1_{\left(y^{(i)} = y_r\right)} \right)$$

• Multinomial Distribution. Pg-8

$P(\vec{x} \mid y_c)$: $\vec{x} = (x_1, x_2, \cdots, x_m)$

Parameters: $\ell, \mu_{1c}, \mu_{2c}, \cdots, \mu_{mc}$
(for $P(\vec{x} \mid y_c)$)

constraint : $\sum_{i=1}^{m} x_i = \ell \qquad x_i \in \mathbb{N}.$

$$P(\vec{x} \mid y_c ; \ell, \vec{\mu_c}) = \frac{n!}{x_1! x_2! \cdots x_m!} \prod_{j=1}^{m} \mu_{jc}^{x_j}$$

where $\vec{\mu_c} = (\mu_{1c}, \mu_{2c}, \cdots, \mu_{mc})$ (for a fixed $y^{(i)} = y_c$)

So,

$$\mu_{jr} = \frac{\left(\sum_{i=1}^{n} 1_{(y^{(i)} = y_r)} x_j^{(i)} \right) + c}{\left(\sum_{i=1}^{n} 1_{(y^{(i)} = y_r)} \left(\sum_{j=1}^{m} x_j^{(i)} \right) \right) + mc}$$

↳ '(c=1)' Laplace Smoothing.

• Gaussian / Normal Distribution.

Parameters : μ_{jc} , σ_{jc}^2 (for $P(x_j | y_c)$)

$x_j \in \mathbb{R}$.

$$P\left(x_j \mid y_c ; \mu_{jc}, \sigma_{jc}^2\right) = \frac{1}{\sqrt{2\pi}\,\sigma_{jc}} e^{-\frac{1}{2}\left(\frac{x_j - \mu_{jc}}{\sigma_{jc}}\right)^2}$$

$$\mu_{jr} = \frac{\sum_{i=1}^{n} 1_{(y^{(i)} = y_r)}\, x_j^{(i)}}{n_r}$$

$$\sigma_{jr}^2 = \frac{\sum_{i=1}^{n} 1_{(y^{(i)} = y_r)} \left(x_j^{(i)} - \mu_{jr}\right)^2}{n_r}$$

$$n_r = \sum_{i=1}^{n} 1_{(y^{(i)} = y_r)}$$

- **Generating output:**

So, now the question remains →

Now that we know the parameters $(\hat{\vec{w}})$ (estimated weights) for any $\vec{x}$ (an input) and any class 'c' amongst the 'k' classes.

$$y = \underset{y_c}{\operatorname{argmax}} \; P(y_c \mid \vec{x}; \hat{\vec{w}})$$

$P(y_c \mid \vec{x}; \hat{\vec{w}})$ → Posterior.

$$= \underset{y_c}{\operatorname{argmax}} \; \frac{P(\vec{x} \mid y_c; \hat{\vec{w}}) \; P(y_c; \hat{\vec{w}})}{P(\vec{x}; \hat{\vec{w}})}$$

$P(\vec{x} \mid y_c; \hat{\vec{w}})$ → likelihood; $P(y_c; \hat{\vec{w}})$ → prior; $P(\vec{x}; \hat{\vec{w}})$ → evidence.

We can ignore the denominator (because it is constant, as input is fixed).

$$= \underset{y_c}{\operatorname{argmax}} \; P(\vec{x} \mid y_c; \hat{\vec{w}}) \cdot P(y_c; \hat{\vec{w}})$$

Simplifying this by Naive Bayes Assumption →

$$y = \underset{y_c}{\operatorname{argmax}} \; P(y_c; \hat{\vec{w}}) \prod_{j=1}^{m} P(x_j \mid y_c; \hat{\vec{w}})$$

We can argmax the 'log' instead (as the product might become very small).

$$y = \underset{y_c}{\operatorname{argmax}} \log\left(P(y_c;\hat{w})\right) + \sum_{j=1}^{m} \log\left(P(x_j \mid y_c;\hat{w})\right)$$

So, for any example $\vec{x}$, the NB classifier outputs the class with maximum value amongst $\rightarrow \left(f(y_1), \dots f(y_k)\right)$

where $f(y_c) = \log\left(P(y_c;\hat{w})\right) + \sum_{j=1}^{m} \log\left(P(x_j \mid y_c;\hat{w})\right)$

Note this is NOT probability.

We are looking at the maximum in the 'log-space'.

$$P\left(y_c \mid \vec{x};\hat{w}\right) = \frac{P(y_c;\hat{w})\, P(\vec{x} \mid y_c;\hat{w})}{P(\vec{x};\hat{w})}$$