Naive Bayes' classifier estimation step

$$
\begin{aligned}
\hat{\boldsymbol{\phi}} &= \underset{\boldsymbol{\phi}}{\operatorname{argmax}} \ \log P(\mathbf{x}, y; \boldsymbol{\mu}, \boldsymbol{\phi}) \\
&= \underset{\boldsymbol{\phi}}{\operatorname{argmax}} \ \log P(\mathbf{x} \,|\, y; \boldsymbol{\mu}, \boldsymbol{\phi}) + \log P(y; \boldsymbol{\mu}) \\
&= \underset{\boldsymbol{\phi}}{\operatorname{argmax}} \ \log P(\mathbf{x} \,|\, y; \boldsymbol{\mu}, \boldsymbol{\phi}) \\
&= \underset{\boldsymbol{\phi}}{\operatorname{argmax}} \ \sum_{j=1}^{N} x_j \log \phi_{y,j}
\end{aligned}
$$

Proof

$$s.t. \quad \sum_{j=1}^{N} \phi_{y,j} = 1 \; \forall y$$

We can use Lagrange's method

$$\mathscr{L}(\phi_y) = \sum_{i\,:\,y^{(i)}=y}^{M} \sum_{j=1}^{N} x_j^{(i)} \log \phi_{y,j} - \lambda\Big(\sum_{j=1}^{N} \phi_{y,j} - 1\Big)$$

where $\mathscr{L}$ is the Lagrangian ← maximize

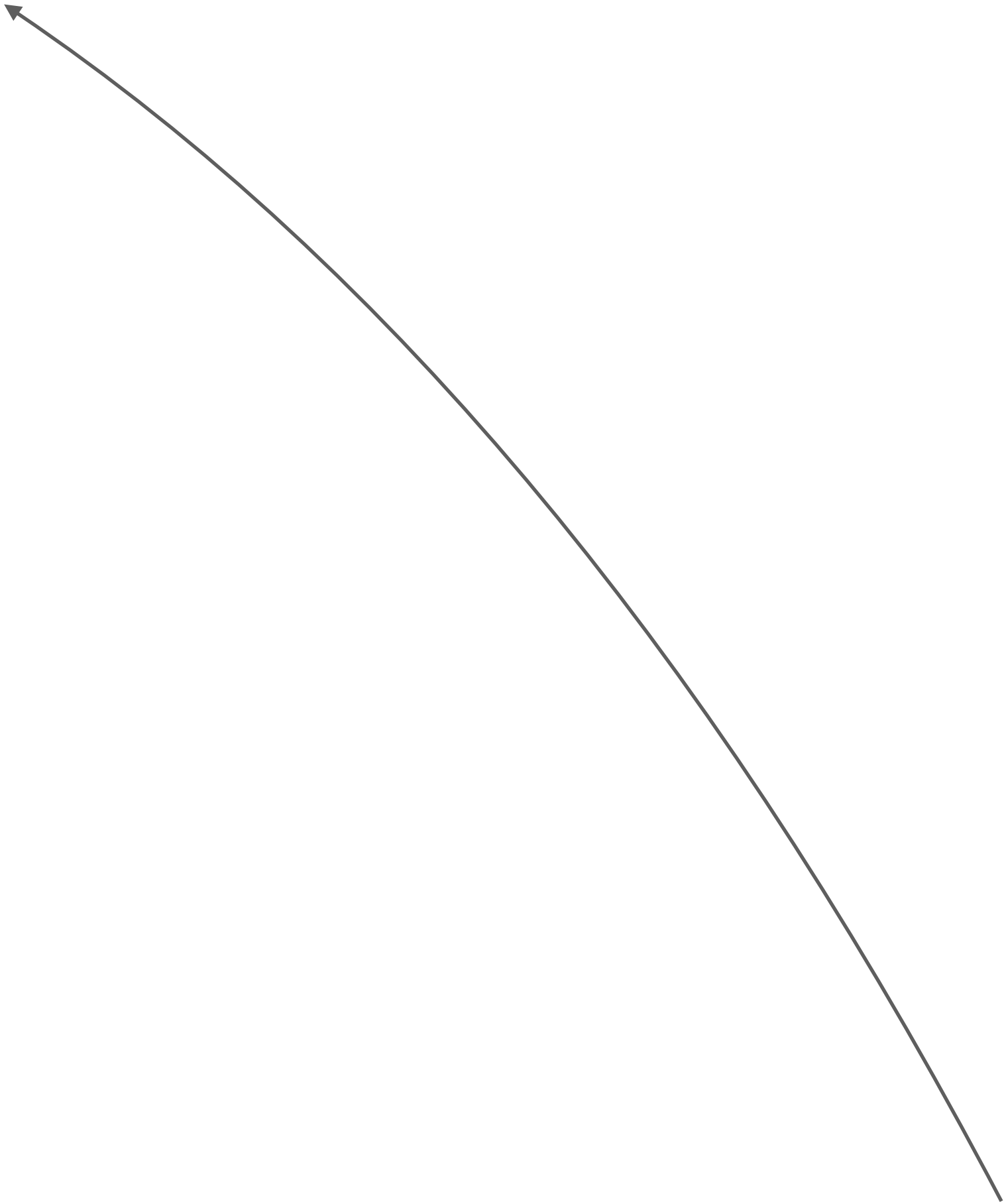
$$\frac{\partial \mathcal{L}(\phi_y)}{\partial \phi_{y,j}} = \sum_{i\,:\,y^{(i)}=y}^{M} \frac{x_j^{(i)}}{\phi_{y,j}} - \lambda \quad \leftarrow \text{set to zero}$$

$$\hat{\phi}_{y,j} \propto \sum_{i\,:\,y^{(i)}=y}^{M} x_j^{(i)}$$

i.e., the ratio between the number of times label *y* appears in conjunction with word x_j, and the number of times label *y* appears in total.

$$\hat{\phi}_{y,j} = \frac{\sum_{i\,:\,y^{(i)}=y}^{M} x_j^{(i)}}{\sum_{j'=1}^{N} \sum_{i\,:\,y^{(i)}=y}^{M} x_{j'}}$$

Maximum likelihood estimate of ϕ_y



Naive Bayes' classifier estimation step

Maximum likelihood estimate of ϕ_y

$$\hat{\phi}_{y,j} = \frac{\sum_{i\,:\,y^{(i)}=y}^{M} x_j^{(i)}}{\sum_{j'=1}^{N}\sum_{i\,:\,y^{(i)}=y}^{M} x_{j'}}$$

i.e., the ratio between the number of times label *y* appears in conjunction with word x_j, and the number of times label *y* appears in total.

Proof

$$\begin{aligned}
\hat{\boldsymbol{\phi}} &= \underset{\boldsymbol{\phi}}{\operatorname{argmax}}\ \log P(\mathbf{x}, y; \boldsymbol{\mu}, \boldsymbol{\phi}) \\
&= \underset{\boldsymbol{\phi}}{\operatorname{argmax}}\ \log P(\mathbf{x} \mid y; \boldsymbol{\mu}, \boldsymbol{\phi}) + \log P(y; \boldsymbol{\mu}) \\
&= \underset{\boldsymbol{\phi}}{\operatorname{argmax}}\ \log P(\mathbf{x} \mid y; \boldsymbol{\mu}, \boldsymbol{\phi}) \\
&= \underset{\boldsymbol{\phi}}{\operatorname{argmax}}\ \sum_{j=1}^{N} x_j \log \phi_{y,j} \\
&\quad s.t. \quad \sum_{j=1}^{N} \phi_{y,j} = 1\ \forall y
\end{aligned}$$

We can use Lagrange's method

$$\mathcal{L}(\phi_y) = \sum_{i\,:\,y^{(i)}=y}^{M}\sum_{j=1}^{N} x_j^{(i)} \log \phi_{y,j} - \lambda\Big(\sum_{j=1}^{N} \phi_{y,j} - 1\Big)$$

where $\mathcal{L}$ is the Lagrangian ← maximize

$$\frac{\partial \mathcal{L}(\phi_y)}{\partial \phi_{y,j}} = \sum_{i\,:\,y^{(i)}=y}^{M} \frac{x_j^{(i)}}{\phi_{y,j}} - \lambda \quad \leftarrow \text{set to zero}$$

$$\hat{\phi}_{y,j} \propto \sum_{i\,:\,y^{(i)}=y}^{M} x_j^{(i)}$$

Smoothing

- One drawback of word counting is that the number of parameters to estimate in the joint distribution over words and labels grows quadratically with the size of the vocabulary and label set. Thus many word-label pairs are never seen in the training set (Zipf's law makes this scaling even worse). One way to address this issue is via smoothing; which reduces the estimators variance at the expense of increasing its bias.

$$\hat{\phi}_{y,j} = \frac{\alpha + \sum_{i\,:\,y^{(i)}=y}^{M} x_j^{(i)}}{N\alpha + \sum_{j'=1}^{N} \sum_{i\,:\,y^{(i)}=y}^{M} x_{j'}}$$