\[ \text{P}(x | y), \quad \text{P}(y) \]

**Maximum Likelihood Estimation**

Find parameters \( \text{P}(x | y) \) and \( \text{P}(y) \) that maximize the probability of observing the training set.

**MAP rule:** Once parameters are discovered, on a new example \((x)\):

\[
\text{P}(y = 0 | x) = \frac{\text{P}(x | y = 0) \cdot \text{P}(y = 0)}{\text{P}(x | y = 0) \cdot \text{P}(y = 0) + \text{P}(x | y = 1) \cdot \text{P}(y = 1)}
\]

\[
\text{P}(y = 1 | x) = \frac{\text{P}(x | y = 1) \cdot \text{P}(y = 1)}{\text{P}(x | y = 1) \cdot \text{P}(y = 1)}
\]

Output 1 if \( \text{P}(y = 1 | x) > \text{P}(y = 0 | x) \)

0 otherwise

**Gaussian Discriminant Analysis**

\[ \text{P}(x | y) \sim N(M_y, \Sigma) \quad \text{and} \quad \text{P}(y) \sim \text{Ber}(\phi) \]

**MAP rule:**

\[
\text{log} \left( \frac{\text{P}(y = 1 | x)}{\text{P}(y = 0 | x)} \right) > 0
\]
\[
\log \frac{p(y=0|x)}{p(y=1|x)} = \theta^T x
\]

\[p(y=1|x) = \frac{1}{1 + e^{-\theta^T x}}\]

**Naive Bayes:**

**Classification:** Given an email, classify if it is spam.

**Input features:** Given email

\[
\begin{bmatrix}
0 \\
\end{bmatrix}
\]

One position for every word in some vocabulary

0 if this word does not appear in the email

1 if this word appears in email

**Discriminative Learning Algorithm**

\[p(y=1|x) \sim \text{Learn} \quad \leadsto \quad \text{Ber}(\phi_x)\]

# parameters is \(2^d \text{ (} x \in \mathbb{R}^d\text{)}\)
Generative Learning Algorithm

Parameter: \( p(y=1) = \Phi_y \)
\( p(x_j \mid y=0) \)
\( p(x_j \mid y=1) \)

Naïve Bayes assumption:
Probability that the \( j \)-th word appears in an email (spam/non-spam) is independent of the \( k \)-th word appearing in an email.

\[
p(x \mid y = 0) = \prod_{j=1}^{d} p(x_j = 1 \mid y = 0)
\]

\[
p(x_j = 1 \mid y = 0) = \Phi_{j,0} \quad \text{-- 2d}
\]

\[
p(x_j = 1 \mid y = 1) = \Phi_{j,1}
\]

Parameters: \( p(y = 1) = \Phi_y \)
\( \Phi_{j,0} \)
\( \Phi_{j,1} \)

\[
L(\Phi_y, \Phi_{j,0}, \Phi_{j,1}) = \prod_{i=1}^{n} \prod_{j=1}^{d} \Phi_{j,y}^{\mathbb{1}[x_{j,i} = 1]} (1 - \Phi_{j,y})^{\mathbb{1}[x_{j,i} = 0]} \]

\[
\mathbb{1}[y = 1] \left[ 1 - \Phi_y \right]^{\mathbb{1}[y = 0]}
\]
We will maximize the log of the likelihood:

\[ \Phi_y = \sum_{i=1}^{n} 1[y^{(i)} = 1] \]

\[ \Phi_j | y = 0 = \sum_{i=1}^{n} 1[x_j^{(i)} = 1 \land y^{(i)} = 0] \leq \sum_{i=1}^{n} 1[y^{(i)} = 0] \leq \sum_{i=1}^{n} 1[y^{(i)} = 1] \]

\[ \Phi_j | y = 1 = \sum_{i=1}^{n} 1[x_j^{(i)} = 1 \land y^{(i)} = 1] \leq \sum_{i=1}^{n} 1[y^{(i)} = 1] \]

\[ \Phi_j = \text{cost}_{j\mid 0} = 0 \]

\[ \Phi_j = \text{cost}_{j\mid 1} = 0 \]

\[ P(y = 1 | x) = \frac{p(x | y = 1) \cdot p(y = 1)}{p(x | y = 0) \cdot p(y = 0) + p(x | y = 1) \cdot p(y = 1)} \]

\[ \text{cost} = \frac{\sum_{j=1}^{m} \Phi_j}{\Phi_y} \]

\[ = 0 \]
Random Variable $Z \sim \text{Multinomial}(k)$

$Z \in \{1, 2, \ldots, k\}$, $\phi_i = P[Z=i]$  

$$\phi_j = 1 + \frac{\sum_{i=1}^k 1[Z=i]}{k+n}$$

Back to email classification

$$\Phi_j | y=s = \frac{1 + \sum 1[\alpha_s^i y^{(i)} = 1 \land y^{(i)} = s]}{2 + \sum 1[y^{(i)} = s]}$$

Generalization:

Goal of learning: Discover a model of the world that explains data.

Minimizing training error is a way to try to ensure that our learnt model will do well on new test data.

Assuming: There is distribution $D$ and training set is generated by sampling from $D$ and new examples
will come from D.
Real goal is minimize
\[ \mathbb{E}_{x \sim D} [\text{loss}] \]

\[ \begin{align*}
\text{Linear} & \quad \text{High Train error} \\
\text{Regression} & \quad \text{High Bias} \quad \text{Underfitting} \\
& \quad \text{High train error} + \text{test error}
\end{align*} \]

Cannot addressed by more training data

Bias / Variance

\[ \begin{align*}
\text{High Poly} & \quad \text{Lowest Training Error} \\
& \quad \text{High Variance} \quad \text{Overfitting} \\
& \quad \text{Low training error} \quad \text{High test error} \quad \text{Addressed increasing training data}
\end{align*} \]
Complexity of mood