Bayes Classifiers

Training time

Joint model \( p(y, x) = p(y) p(x | y) \)
\[ p(y=k) = \frac{\# k \text{ labels}}{N} \]
\[ p(x | y=k) \ldots \text{eg } N(x; \mu^{(k)}, \Sigma^{(k)}) \]

\[ \text{Not Bayesian } \]
\[ \text{Assuming we know all parameters.} \]

Naive Bayes \( p(x | y=k) = \prod_d p(x_d | y=k) \)

Test time

\[ p(y | x) \propto p(y, x) \] (Bayes' Rule)

\[ y_{\text{guess}} = \text{argmax}_k p(y=k, x) \] 

"goodness"
Regression to labels

\[ f(x) = w^T x + b \]

If \( f(x) > \frac{1}{2} \), guess \( y = 1 \)

\[ f(x) = w_1 + w_2 x + w_3 x^2 \]

Fit with RBFs
If minimize square loss?

Minimize \[ E \left[ (y - f(x))^2 \right] \] at some location \( x \)

\[
\text{Cost} = p_1 (1-f)^2 + (1-p_1) (0-f)^2
\]

\[
= \sum_{p(y=1|x)} p_1 p(y=1|x) \quad \sum_{p(y=0|x)} (1-p_1) p(y=0|x)
\]

\[
= p_1 (1-2f + f^2) + (1-p_1) f^2
\]

\[
= f^2 (p_1 - p_1^2) - 2 p_1 f + p_1
\]

\[
\frac{\partial \text{cost}}{\partial f} = 2f - 2 p_1 = 0 \text{ at optimum}
\]

\[
f = p_1
\]
Multiple classes

$y \in \{1, 2, 3, 4 \ldots 10\}$

"sport" \rightarrow "romance"

"crime"

$\mathbf{f}(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$

Maybe replace $\mathbf{x}$ with $\phi(\mathbf{x})$

$\mathbf{f}(\mathbf{x}^{(1)}) \approx 1 \rightarrow \text{"sport"}$

$\mathbf{f}(\mathbf{x}^{(2)}) \approx 3 \rightarrow \text{"romance"}$

$\mathbf{f}(\frac{\mathbf{x}^{(1)} + \mathbf{x}^{(2)}}{2}) \approx 2 \rightarrow \text{"crime"}$
One-hot encoding, One-of-K-encoding

Vector output

\[ y^{(n)} = [0 \ 0 \ 0 \ \ldots \ 0 \ 1 \ 0 \ \ldots \ 0]^T \]

Is in \( k^{th} \) position

Kx1 vector

If we have \( K \) classes

If \( n^{th} \) example is in class \( k \)

Fit \( K \) functions, one for each bit \( y_k \)

This pre-processing step is also useful for input features

\( x_n \in \{ \text{"red"}, \text{"green"}, \text{"blue"} \} \)

\( \{ 1, 2, 3 \} \)

3 features

\begin{align*}
\text{red} & \rightarrow 1000 \\
\text{green} & \rightarrow 0100 \\
\text{blue} & \rightarrow 0010
\end{align*}

R doesn't create this column.

Puzzle: In R you can do one-hot encoding

\begin{align*}
\text{red} & \rightarrow 10 \\
\text{green} & \rightarrow 01 \\
\text{blue} & \rightarrow 00
\end{align*}
Gradients for least squares cost

Residuals: \( r = y - Xw \)

\( N \times 1 \) vector of scalar labels

Cost: \( r^T r = (y - Xw)^T (y - Xw) \)

\[ = y^T y - 2w^T (X^T y) + w^T X^T X w \]

"Gradient" vector of partial derivatives:

\[ \nabla_w [r^T r] = -2(X^T y) + 2X^T X w \]
\[ \nabla_w [w^T h] = \begin{bmatrix} \frac{\partial w^T h}{\partial w_1} \\ \frac{\partial w^T h}{\partial w_2} \\ \vdots \\ \frac{\partial w^T h}{\partial w_D} \end{bmatrix} = h \]

\[ \frac{\partial w^T h}{\partial w_i} = \frac{\partial}{\partial w_i} \sum_j w_j h_j = \frac{\partial}{\partial w_i} \left( w_i h_i + w_2 h_2 + \ldots \right) \]

\[ = h_i \]

Matrix Cookbook.