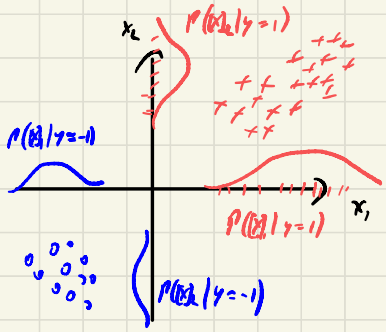
Naive Bayes

Assumption: $P(\vec{x}|y) = \prod_{\alpha=1}^{d} P([\vec{x}]_\alpha | y)$

Bayes' Rule $\Rightarrow$ $P(y|\vec{x}) = \frac{\left[\prod_{\alpha=1}^{d} P([\vec{x}]_\alpha|y)\right] P(y)}{\sum_{y'} \prod_{\alpha=1}^{d} P([\vec{x}]_\alpha|y') P(y')}$ ($P(\vec{x}|y)$)

Gaussian NB

Each feature is modeled with a Gaussian.



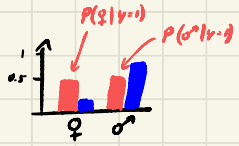
Categorical NB

Each feature falls into categories and is modeled by its own multinomial distribution.

$[\vec{x}]_1 \in \{\text{male}, \text{female}\}$

$[\vec{x}]_2 \in \{NY, MA, PA\}$

$[\vec{x}]_3 \in \{\text{fever}, \text{no fever}\}$



Multinomial NB:

Each feature α denotes the count how often a d-sided die resulted in side α.

one die for each class $\begin{cases} P([\vec{x}]_\alpha|y=1) \\ P([\vec{x}]_\alpha|y=-1) \end{cases}$

e.g. features are 1...6 you roll a 6-sided die 10 times $\{1, 1, 5, 3, 1, 2, 5, 6, 1, 2\}$

Example: Spam filter / text classification $\Rightarrow$ $\vec{x} = \begin{pmatrix} 4 \\ 2 \\ 1 \\ 0 \\ 2 \\ 1 \end{pmatrix} \begin{matrix} \leftarrow 1 \\ \leftarrow 2 \\ \leftarrow 3 \\ \leftarrow 4 \\ \leftarrow 5 \\ \leftarrow 6 \end{matrix}$

$P([\vec{x}]_\alpha = 4|y=-1) = (\theta_{\alpha,-1})^4$ $\quad$ $P([\vec{x}]_\alpha = 4|y=+1) = (\theta_{\alpha,+1})^4$

$$\log\left[P(\vec{x}|y=-1)\right] = \log\left[\prod_{\alpha=1}^{d} (\theta_{\alpha,-1})^{x_\alpha}\right] = \sum_{\alpha=1}^{d} x_\alpha \underbrace{\log[\theta_{\alpha,-1}]}_{u_\alpha^-} = \sum_{\alpha=1}^{d} x_\alpha u_\alpha^- = \vec{x}^T \vec{u}^- \Rightarrow P(\vec{x}|y=-1) = e^{\vec{x}^T\vec{u}^- + b^-}$$

$$\Rightarrow P(y=+1|\vec{x}) = \frac{P(\vec{x}|y=+1)P(y=+1)}{P(\vec{x}|y=+1)P(y=+1) + P(\vec{x}|y=-1)P(y=-1)} = \frac{e^{\vec{x}^T\vec{u}^+ + b^+}}{e^{\vec{x}^T\vec{u}^+ + b^+} + e^{\vec{x}^T\vec{u}^- + b^-}} = \frac{1}{1 + e^{-(\vec{x}^T\vec{u} + b)}}$$

$u = u^+ - u^-$
$b = b^+ - b^-$

$b^- = \log P(y=-1)$

Naive Bayes is a linear classifier

1. Suppose that $y_i \in \{-1, +1\}$ and features are multinomial. We can show that

$$P(y \mid \mathbf{x}) = \frac{1}{1 + e^{-y(\mathbf{w}^\top \mathbf{x} + b)}}.$$

As before, we define $P(x_\alpha | Y = +1) \propto \theta_{\alpha+}^{x_\alpha}$. Let us define two weight vectors $\mathbf{w}_+, \mathbf{w}_-$ and bias terms b^+, b^- as $w_\alpha^+ = \log[\theta_{\alpha+}]$ and $b^+ = \log[P(Y = +1)]$. This notation allows us to define the following relation:

$$\begin{aligned}
\log\left[P(\mathbf{x}|Y = +1)\right] &= \log\left[\Pi_{\alpha=1}^{d} P(x_\alpha \mid Y = +1)\right] \\
&= \sum_{\alpha=1}^{d} x_\alpha \log[\theta_{\alpha+}] \\
&= \sum_{\alpha=1}^{d} x_\alpha w_\alpha^+ \\
&= \mathbf{x}^\top \mathbf{w}_+.
\end{aligned}$$

It follows that $P(\mathbf{x}|Y = +1) = e^{\mathbf{x}^\top \mathbf{w}_+}$ and $P(\mathbf{x}|Y = -1) = e^{\mathbf{x}^\top \mathbf{w}_-}$. Also, by definition $P(Y = +1) = e^{b^+}$ and $P(Y = -1) = e^{b^-}$. Let us define the differences between the two weight vectors and biases as: $\mathbf{w} = \mathbf{w}_- - \mathbf{w}_+$ and $b = b_- - b_+$. We can use Bayes Rule to derive:

$$\begin{aligned}
P(Y = +1 \mid \mathbf{x}) &= \frac{P(\mathbf{x} \mid +1) P(Y = +1)}{P(\mathbf{x} \mid +1) P(Y = +1) + P(\mathbf{x} \mid -1) P(Y = -1)} \\
&= \frac{e^{\mathbf{x}^\top \mathbf{w}_+ + b_+}}{e^{\mathbf{x}^\top \mathbf{w}_+ + b_+} + e^{\mathbf{x}^\top \mathbf{w}_- + b_-}} \\
&= \frac{1}{1 + e^{-(\mathbf{x}^\top \mathbf{w} + b)}}
\end{aligned}$$

Finally, because our labels $y \in \{+1, -1\}$ we can conveniently create one equation for both classes:

$$P(Y = y \mid \mathbf{x}) = \frac{1}{1 + e^{-y(\mathbf{x}^\top \mathbf{w} + b)}}.$$

Logistic Regression

previous next

In this lecture we will learn about the discriminative counterpart to the Gaussian Naive Bayes (Naive Bayes for continuous features).

Machine learning algorithms can be (roughly) categorized into two categories:

- *Generative* algorithms, that estimate $P(\mathbf{x}_i, y)$ (often they model $P(\mathbf{x}_i|y)$ and $P(y)$ separately).
- *Discriminative* algorithms, that model $P(y|\mathbf{x}_i)$

The Naive Bayes algorithm is *generative*. It models $P(\mathbf{x}_i|y)$ and makes explicit assumptions on its distribution (e.g. multinomial, categorical, Gaussian, ...). The parameters of this distributions are estimated with MLE or MAP. We showed previously that for the Gaussian Naive Bayes $P(y|\mathbf{x}_i) = \frac{1}{1+e^{-y(\mathbf{w}^T\mathbf{x}_i+b)}}$ for $y \in \{+1, -1\}$ for specific vectors $\mathbf{w}$ and b that are uniquely determined through the particular choice of $P(\mathbf{x}_i|y)$.

Logistic Regression is often referred to as the *discriminative* counterpart of Naive Bayes. Here, we model $P(y|\mathbf{x}_i)$ and assume that it takes on exactly this form

$$P(y|\mathbf{x}_i) = \frac{1}{1 + e^{-y(\mathbf{w}^T\mathbf{x}_i+b)}}.$$

We make little assumptions on $P(\mathbf{x}_i|y)$, e.g. it could be Gaussian or Multinomial. Ultimately it doesn't matter, because we estimate the vector $\mathbf{w}$ and b directly with MLE or MAP to maximize the conditional likelihood of $\Pi_i P(y_i|\mathbf{x}_i; \mathbf{w}, b)$. For a lot more details, I strongly suggest that you read this excellent book chapter by Tom Mitchell.

Throughout this lecture we absorbed the parameter b into $\mathbf{w}$ through an additional constant dimension (similar to the Perceptron).

Maximum likelihood estimate (MLE)

In MLE we choose parameters that **maximize the conditional likelihood**. The conditional data likelihood $P(\mathbf{y} \mid X, \mathbf{w})$ is the probability of the observed values $\mathbf{y} \in \mathbb{R}^n$ in the training data conditioned on the feature values $\mathbf{x}_i$. Note that

$X = [\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$. We choose the paramters that maximize this function and we assume that the y_i's are independent given the input features $\mathbf{x}_i$ and $\mathbf{w}$. So,

$$
\begin{aligned}
\hat{\mathbf{w}}_{MLE} &= \underset{\mathbf{w}}{\operatorname{argmax}}\, P(D|\mathbf{w}) && \text{(Definition of MLE)} \\
&= \underset{\mathbf{w}}{\operatorname{argmax}}\, P((y_1, \mathbf{x}_1), \ldots, (y_n, \mathbf{x}_n) \mid \mathbf{w}) && \text{(Substituting in D.)} \\
&= \underset{\mathbf{w}}{\operatorname{argmax}} \prod_{i=1}^{n} P(y_i, \mathbf{x}_i \mid \mathbf{w}) && \text{(Data is i.i.d.)} \\
&= \underset{\mathbf{w}}{\operatorname{argmax}} \prod_{i=1}^{n} P(y_i \mid \mathbf{x}_i, \mathbf{w}) P(\mathbf{x}_i \mid \mathbf{w}) && \text{(Chain Rule of Statistics)} \\
&= \underset{\mathbf{w}}{\operatorname{argmax}} \prod_{i=1}^{n} P(y_i \mid \mathbf{x}_i, \mathbf{w}) P(\mathbf{x}_i) && (\mathbf{x}_i \text{ does not depend on } \mathbf{w}) \\
&= \underset{\mathbf{w}}{\operatorname{argmax}} \prod_{i=1}^{n} P(y_i \mid \mathbf{x}_i, \mathbf{w}) && (P(\mathbf{x}_i) \text{ does not affect } \mathbf{w}) \\
&= \underset{\mathbf{w}}{\operatorname{argmax}} \sum_{i=1}^{n} \log\left[P(y_i \mid \mathbf{x}_i, \mathbf{w})\right]. && \text{(Taking the } \mathbf{log}) \\
&= \underset{\mathbf{w}}{\operatorname{argmax}} -\sum_{i=1}^{n} \log(1 + e^{-y_i \mathbf{w}^T \mathbf{x}_i}) && \text{(Substituting in } P(y_i \mid \mathbf{x}_i, \mathbf{w})) \\
&= \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{n} \log(1 + e^{-y_i \mathbf{w}^T \mathbf{x}_i}) && \text{(We prefer minimization.)}
\end{aligned}
$$

We need to estimate the parameters $\mathbf{w}$. To find the values of the parameters at minimum, we can try to find solutions for $\nabla_{\mathbf{w}} \sum_{i=1}^{n} \log(1 + e^{-y_i \mathbf{w}^T \mathbf{x}_i}) = 0$. This equation has no closed form solution, so we will use Gradient Descent on the *negative log likelihood* $\ell(\mathbf{w}) = \sum_{i=1}^{n} \log(1 + e^{-y_i \mathbf{w}^T \mathbf{x}_i})$.

Maximum a Posteriori (MAP) Estimate

In the MAP estimate we treat $\mathbf{w}$ as a random variable and can specify a prior belief distribution over it. We may use: $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \sigma^2 I)$. This is the Gaussian approximation for LR.

Our goal in MAP is to find the *most likely* model parameters *given the data*, i.e., the parameters that **maximaize the posterior**.

$$P(\mathbf{w} \mid D) = P(\mathbf{w} \mid X, \mathbf{y}) \propto P(\mathbf{y} \mid X, \mathbf{w})\, P(\mathbf{w}),$$

We can solve for $\hat{\mathbf{w}}_{MA}$ just as before with MLE.

$$
\begin{aligned}
\hat{\mathbf{w}}_{MAP} &= \underset{\mathbf{w}}{\operatorname{argmax}}\, P(D|\mathbf{w})P(\mathbf{w}) && \text{(Definition of MA} \\
&= \underset{\mathbf{w}}{\operatorname{argmax}}\, P((y_1, \mathbf{x}_1), \ldots, (y_n, \mathbf{x}_n) \mid \mathbf{w})P(\mathbf{w}) && \text{(Substituting in} \\
&= \underset{\mathbf{w}}{\operatorname{argmax}} \left(\prod_{i=1}^{n} P(y_i \mid \mathbf{x}_i, \mathbf{w}) \right) P(\mathbf{w}) && \text{(Data is i.i.} \\
&= \underset{\mathbf{w}}{\operatorname{argmax}} \sum_{i=1}^{n} \log \left[P(y_i \mid \mathbf{x}_i, \mathbf{w}) \right] + \log P(\mathbf{w}) && \text{(Take } \mathbf{log} \text{ and sum proper} \\
&= \underset{\mathbf{w}}{\operatorname{argmin}} - \sum_{i=1}^{n} \log \left[P(y_i \mid \mathbf{x}_i, \mathbf{w}) \right] - \log P(\mathbf{w}) && \text{(We prefer minimizatic} \\
&= \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{n} \log \left[1 + e^{-y_i \mathbf{w}^T \mathbf{x}_i} \right] + \frac{1}{2\sigma^2} \mathbf{w}^\top \mathbf{w} && \text{(Substituting } P(y_i \mid \mathbf{x}_i, \mathbf{w}) \text{ and } P(\mathbf{v} \\
&= \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{n} \log \left[1 + e^{-y_i \mathbf{w}^T \mathbf{x}_i} \right] + \lambda \mathbf{w}^\top \mathbf{w} && \text{(Substitute } \lambda = \frac{1}{2\sigma}
\end{aligned}
$$

where $\lambda = \frac{1}{2\sigma^2}$. Once again, this function has no closed form solution, but we can use Gradient Descent on the *negative log posterior* $\ell(\mathbf{w}) = \sum_{i=1}^{n} \log(1 + e^{-y_i \mathbf{w}^T \mathbf{x}_i}) + \lambda \mathbf{w}^\top \mathbf{w}$ to find the optimal parameters $\mathbf{w}$.

For a better understanding for the connection of Naive Bayes and Logistic Regression, you may take a peek at these excellent notes.

Summary

Logistic Regression is the discriminative counterpart to Naive Bayes. In Naive Bayes, we first model $P(\mathbf{x}|y)$ for each label y, and then obtain the decision boundary that best discriminates between these two distributions. In Logistic Regression we do not attempt to model the data distribution $P(\mathbf{x}|y)$, instead, we model $P(y|\mathbf{x})$ directly. We assume the same probabilistic form $P(y|\mathbf{x}_i) = \frac{1}{1+e^{-y(\mathbf{w}^T \mathbf{x}_i + b)}}$, but we do not restrict ourselves in any way by making assumptions about $P(\mathbf{x}|y)$ (in fact it can be any member of the Exponential Family). This allows logistic regression to be more flexible, but such flexibility also requires more data to avoid overfitting. Typically, in scenarios with little data and if the modeling assumption is appropriate, Naive Bayes tends to outperform Logistic Regression. However, as data sets become large logistic regression often outperforms Naive Bayes, which suffers from the fact that the assumptions made on $P(\mathbf{x}|y)$ are probably not exactly correct. If the assumptions hold exactly, i.e. the data is truly drawn from the distribution that we assumed in Naive Bayes, then Logistic Regression and Naive Bayes converge to the exact same result in the limit (but NB will be faster).