Naive Bayes is a *generative model*. It models the joint $p(x, y)$, with the independence assumption on the features of $x$. However, if our ultimate goal is classification, the relevant part is $p(y|x)$. In Naive Bayes this is computed via Bayes rule. One might wonder whether it is possible to estimate $p(y|x)$ directly. A model that estimates $p(y|x)$ directly is known as a *discriminative model*. Logistic regression is one such model.

Consider binary classification with $y = -1, 1$, and each example is represented by a feature vector $x$. The intuition is to first map $x$ to a real number, such that large positive number means that $x$ is likely to be positive ($y = 1$), and negative number means $x$ is negative ($y = -1$). This can be done via the inner product between $x$ and a parameter vector $\theta \in \mathbb{R}^d$:

$$\theta^\top x \quad (1)$$

This is a linear mapping with value in $(-\infty, \infty)$.

A practical note: It is often convenient to append a constant 1 as an extra feature to the $x$ vector, and the dimensionality of $\theta$ is increased by one accordingly. This new dimension serves as an offset, which is equivalent to $\theta^\top x + \theta_0$.

The next step is to “squash” the range down to $[0, 1]$ so one can interpret it as the label probability. This is done via the *logistic function*:

$$p(y = 1|x) = \sigma(\theta^\top x) = \frac{1}{1 + \exp(-\theta^\top x)} \quad (2)$$

For binary classification with -1, 1 class encoding, one can unify the definition for $p(y = 1|x)$ and $p(y = -1|x)$ with

$$p(y|x) = \frac{1}{1 + \exp(-y\theta^\top x)} \quad (3)$$

Logistic regression can be easily generalized to multiple classes. Let there be $K$ classes. Each class has its own parameter $\theta_k$. The probability is defined via the *softmax function*

$$p(y = k|x) = \frac{\exp(\theta_k^\top x)}{\sum_{i=1}^{K} \exp(\theta_i^\top x)} \quad (4)$$

We will focus on binary classification in the rest of this note.

### 0.1 Training

Training (i.e., finding the parameter $\theta$) can be done by maximizing the *conditional* log likelihood of the training data $\{(x, y)_{1:n}\}$:

$$\max_{\theta} \sum_{i=1}^{n} \log p(y_i|x_i, \theta) \quad (5)$$

However, when the training data is linearly separable, two bad things happen: 1. $\|\theta\|$ goes to infinity; 2. There are infinite number of MLE’s. To see this, note any step function (sigmoid with $\|\theta\| = \infty$) that is in the gap between the two classes is an MLE.

---

1Strictly speaking, one needs only $K - 1$ parameter vectors.
Logistic Regression

One way to avoid this is to incorporate a prior on $\theta$ in the form of a zero-mean Gaussian with covariance $\frac{1}{2\lambda}I$,

$$\theta \sim \mathcal{N}(0, \frac{1}{2\lambda}I),$$

and seek the MAP estimate. This is essentially smoothing, since large $\theta$ values will be penalized more. That is, we seek the $\theta$ that maximizes

$$\log p(\theta) + \sum_{i=1}^{n} \log p(y_i|x_i, \theta)$$

$$= -\lambda\|	heta\|^2 + \sum_{i=1}^{n} \log p(y_i|x_i, \theta)$$

$$= -\lambda\|	heta\|^2 - \sum_{i=1}^{n} \log (1 + \exp(-y_i\theta^\top x_i)).$$

Equivalently, one minimizes the $\ell_2$-regularized negative log likelihood loss

$$\min_{\theta} \lambda\|	heta\|^2 + \sum_{i=1}^{n} \log (1 + \exp(-y_i\theta^\top x_i)).$$

This is a convex function so there is a unique global minimum of $\theta$. Unfortunately there is no closed form solution. One typically solves the optimization problem with Newton-Raphson iterations, also known as iterative reweighted least squares for logistic regression.

0.2 Graphical Model

Logistic regression can be represented as a directed graphical model (Bayes Network) with a $y$ node, and a set of $x$ nodes (one for each feature dimension). The arrows go from the $x$ nodes to the $y$ node. Note this is exactly the opposite of Naive Bayes models.

Further notice that we do not model $p(x)$ in logistic regression. Therefore it is not possible to have a sampling program to generate $(x, y)$ data. This is a difference between generative (e.g., Naive Bayes) vs. discriminative (e.g., logistic regression) models.

0.3 Logistic Regression as a Linear Classifier

Logistic regression is a linear classifier, with the decision boundary

$$\theta^\top x = 0.$$