Supervised Learning

Repeating the mistakes of the past

Naive Bayes

Probability

$P(\text{Toss} = \text{heads}) = 0.5.$

Independence

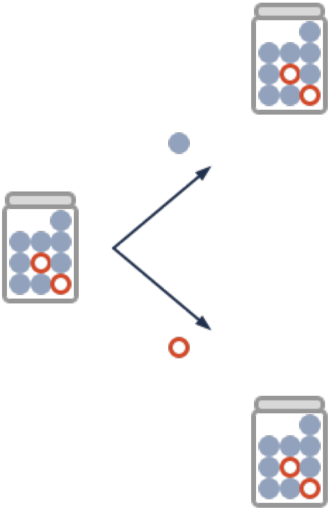
$P(\text{Toss} = \text{heads})$ = 0.5.
$P(\text{Toss} = \text{heads} | \text{LastToss} = \text{heads})$ = 0.5.

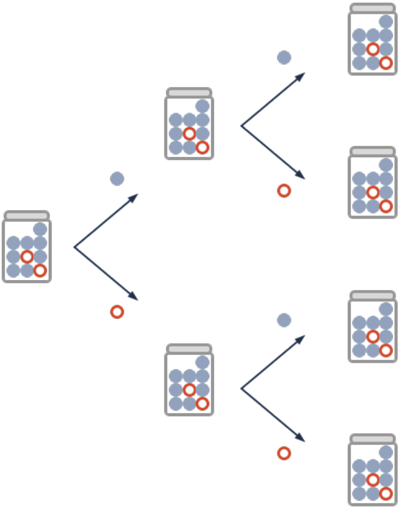
Marbles Problem

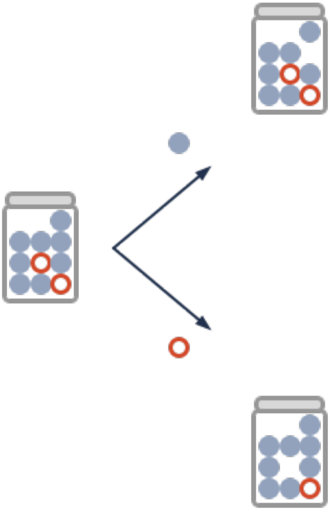
8 Blue, 2 Orange

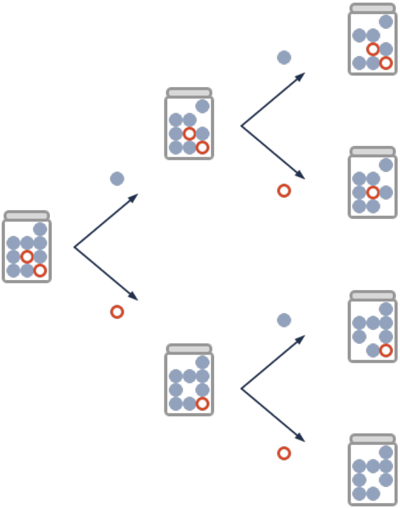
$P(\text{Select} = \text{blue}) = \frac{8}{8+2} = 0.8$

$P(\text{Select} = \text{orange}) = \frac{2}{8+2} = 0.2$









Marbles Problem

8 Blue, 2 Orange
$P(\text{Select} = \text{blue} \mid \text{Previous} = \text{blue})$
If removing marbles, $\frac{7}{7+2} = 0.78$
If returning marbles, $\frac{8}{8+2} = 0.8$

Bayes' Rule

$$P(A \mid B) = \frac{P(A)\; P(B|A)}{P(B)}.$$

Naive Bayes

k classes where each datum has d features

$$P(\text{Class} = i \mid x_1, \ldots, x_d) = \frac{P(\text{Class} = i)\, P(x_1, \ldots, x_d \mid \text{Class} = i)}{P(x_1, \ldots, x_d)}.$$

Naive Bayes

Naive because we assume that features are conditionally independent,
$P(x_\alpha \mid \text{Class} = i, x_1, \ldots, x_{\alpha-1}, x_{\alpha+1}, \ldots, x_d) = P(x_\alpha \mid \text{Class} = i)$.

Naive Bayes

$$P(\text{Class} = i \mid x_1, \ldots, x_d) = \frac{P(\text{Class} = i) \prod_{a=1}^{n} P(x_a \mid \text{Class} = i)}{P(x_1, \ldots, x_d)}.$$

Naive Bayes

$$\text{Class} = \arg\max_i P(\text{Class} = i) \prod_{a=1}^{n} P(x_a \mid \text{Class} = i)$$

Gaussian NB

$$\underbrace{P(x_a \mid \text{Class} = i)}_{\textit{Gaussian}} = \frac{\exp \frac{(x_a - \mu_{a,i})^2}{2\sigma_{a,i}^2}}{\sqrt{2\pi\sigma_{a,i}^2}},$$

where $\sigma_{a,i}$ and $\mu_{a,i}$ are the standard deviation and mean of feature a for class i.

Multinomial NB

$$\underset{\textit{Multinomial}}{P(x_a \mid \text{Class} = i)} = \frac{\sum_{\mathbf{x} \in I} x_a + \alpha}{\sum_{a=1}^{n} \sum_{\mathbf{x} \in I} x_a + \alpha\, n},$$

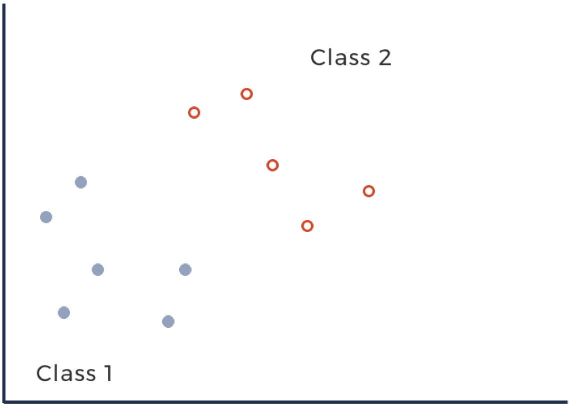
where I is the set of feature vectors of class i, and α is a smoothing parameter.

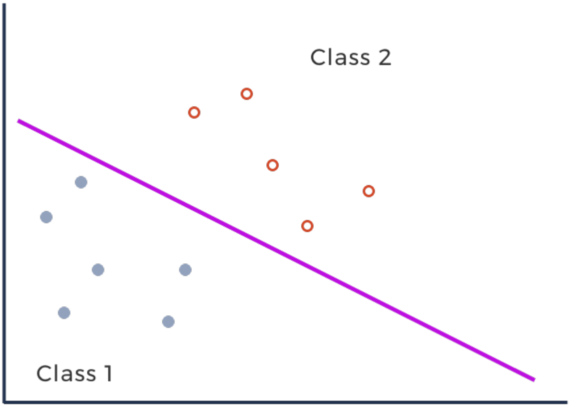
Bernouli NB

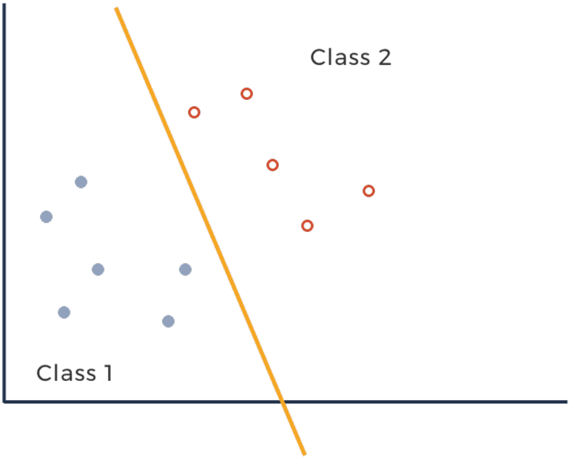
$$P(x_a \underset{Bernouli}{|} \text{Class} = i) = \frac{\sum_{\mathbf{x} \in I} x_a}{m_i} x_a + (1 - \frac{\sum_{\mathbf{x} \in I} x_a}{m_i})(1 - x_a),$$

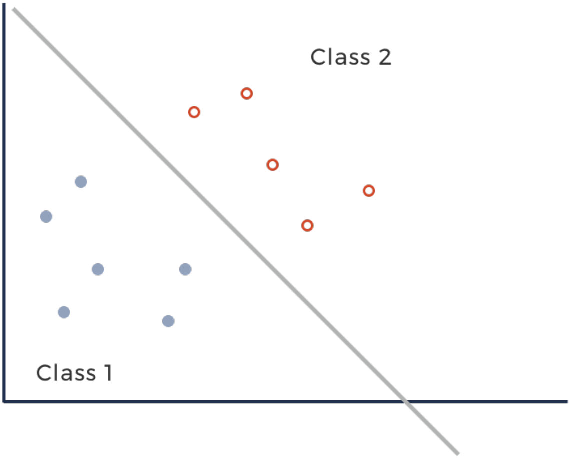
where all n features are boolean values, I is the set of feature vectors of class i, and m_i is the number of vectors in class i.

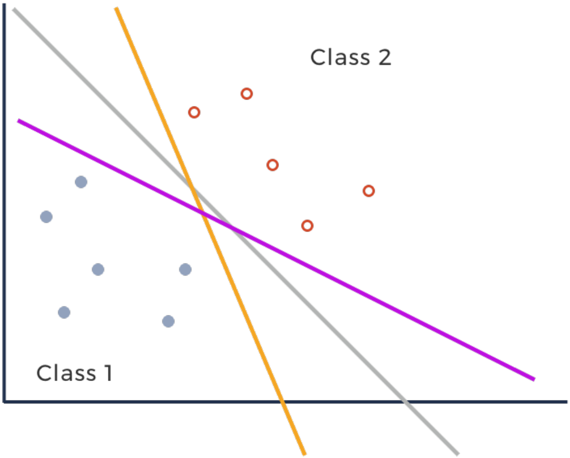
Support Vector Machines

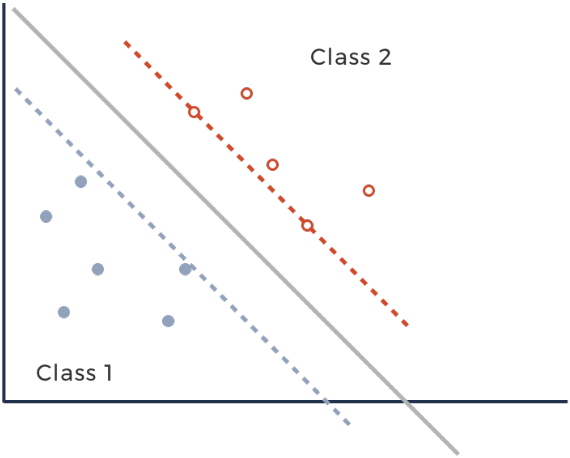

Class 2
Class 1

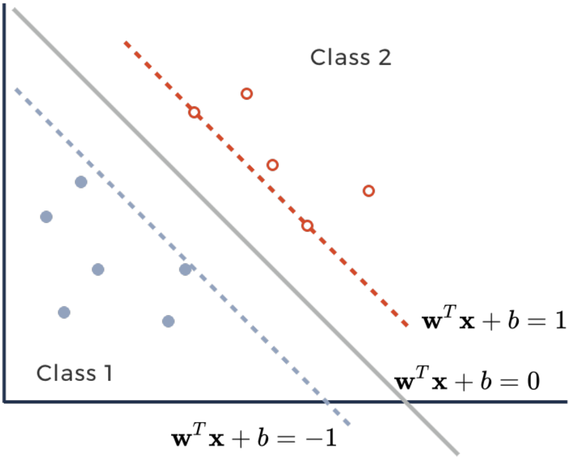

Class 2
Class 1

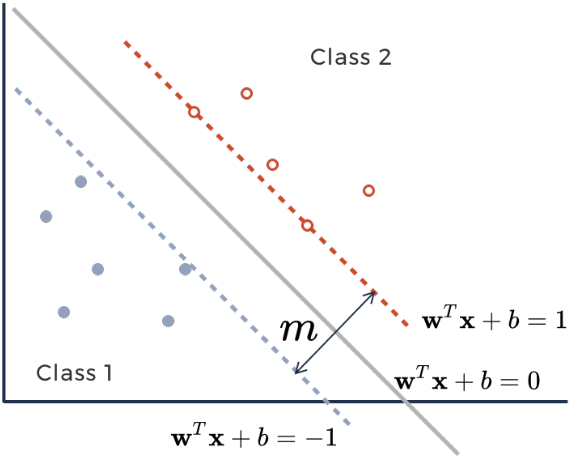

Class 2
Class 1


Class 2
Class 1

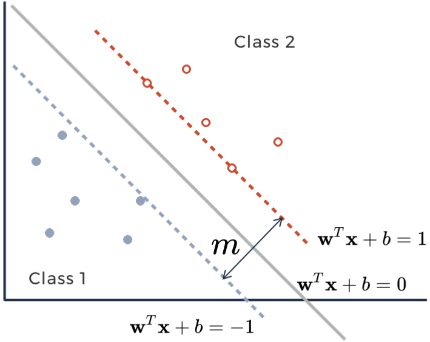

Class 2
Class 1


Class 2
Class 1


Class 2
$\mathbf{w}^T\mathbf{x} + b = 1$
Class 1
$\mathbf{w}^T\mathbf{x} + b = 0$
$\mathbf{w}^T\mathbf{x} + b = -1$


Class 2
$\mathbf{w}^T\mathbf{x} + b = 1$
m
Class 1
$\mathbf{w}^T\mathbf{x} + b = 0$
$\mathbf{w}^T\mathbf{x} + b = -1$

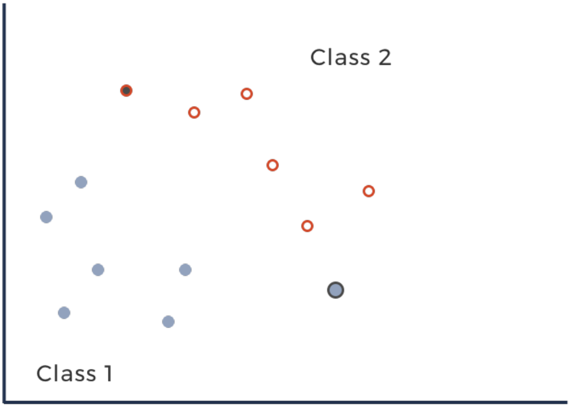
Margin

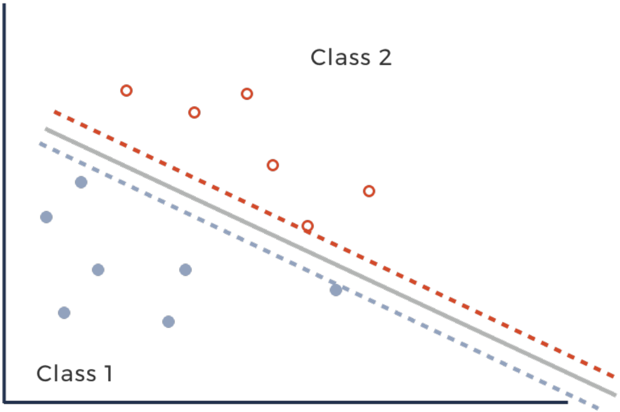


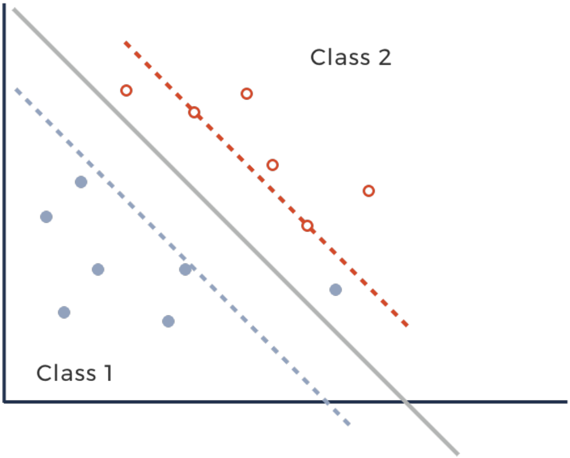
$$\text{margin} = \frac{2}{||\mathbf{w}||}.$$

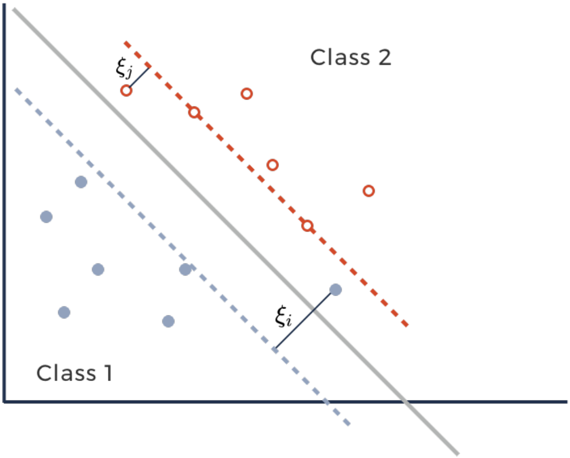
$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2$$
$$s.t. \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \qquad \forall\, i,$$

where $y_i \in [1, -1]$ determines the class point i belongs to.


Class 2
Class 1


Class 2
Class 1


Class 2
Class 1

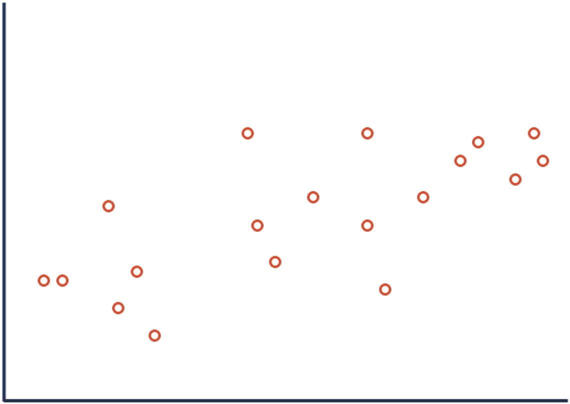

Class 2
ξ_j
ξ_i
Class 1

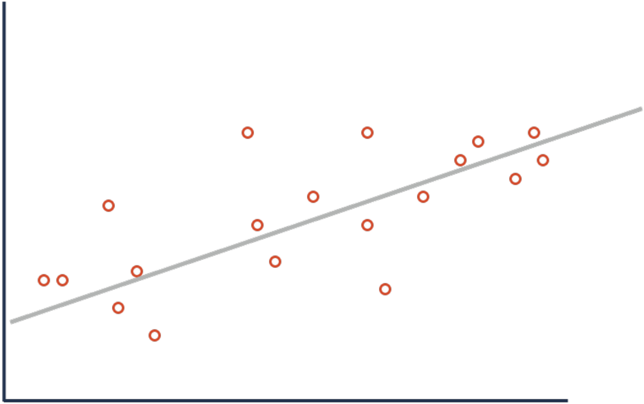
$$
\begin{aligned}
&\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i \\
&s.t. \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \xi_i \geq 0 \qquad \forall\, i,
\end{aligned}
$$

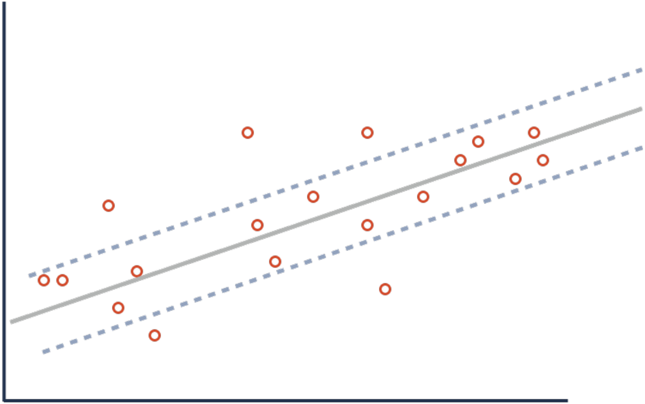
where ξ_i is the error for point i, and C is the penalty tuning parameter.

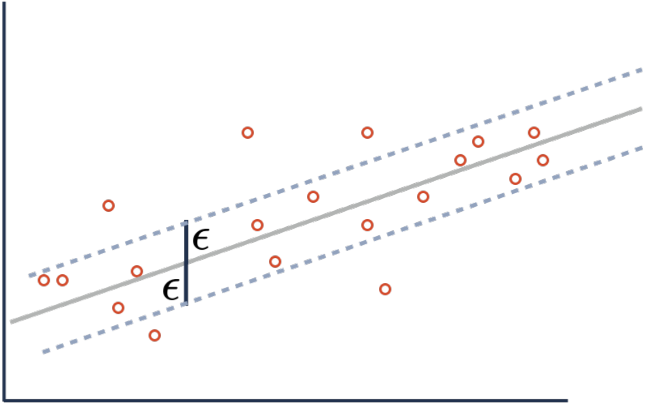
Multi-Class

- One vs All / Rest
- One vs One

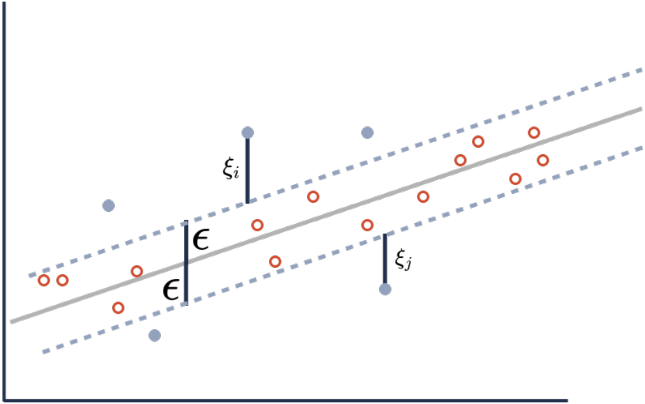








ϵ
ϵ



ξ_i
ϵ
ϵ
ξ_j

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n} \xi_i$$

$$s.t. \quad |y_i - (\langle \mathbf{w}, \mathbf{x}_i \rangle + b)| \leq \epsilon + \xi_i \qquad , \epsilon, \xi_i \geq 0 \; \forall \; i,$$

where ϵ is the acceptable error boundary.

Kernel Method

Transform the data into linearly separable features.

Kernel function

$\phi(\mathbf{x})$ defines a transformation that occurs to any given point into the new space

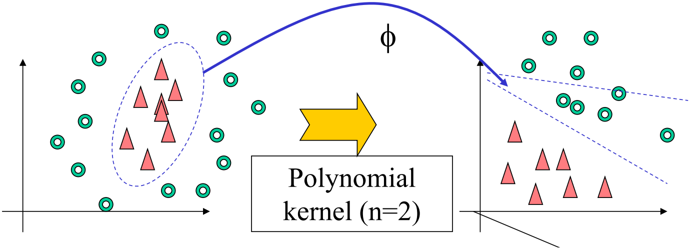
$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j).$$

Polynomial Kernel

$$\underbrace{K(\mathbf{x}_i, \mathbf{x_j})}_{Polynomial} = (\mathbf{x}_i^T \mathbf{x}_j + 1)^d,$$

with degree d,

Polynomial Kernel

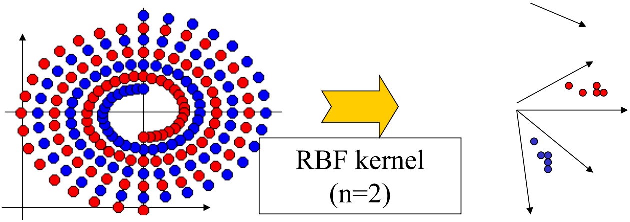


RBF Kernel

$$\underset{RBF}{K(\mathbf{x}_i, \mathbf{x_j})} = \exp(\frac{-||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2}),$$

with width σ,

RBF Kernel



Sigmoid Kernel

$$\underbrace{K(\mathbf{x}_i, \mathbf{x_j})}_{Sigmoid} = \tanh\left(\kappa \mathbf{x}_i^T \mathbf{x}_j + \theta\right),$$

with parameters κ and θ.

Questions