L101: Machine Learning for Language Processing

Lecture 4

Guy Emerson

Today's Lecture

- Decision boundaries
- Non-probabilistic classifiers
  - Support Vector Machine
  - Kernels
- Linguistic kernels

Decision Boundaries

$$\operatorname*{argmax}_{y} P(y|x)$$

Decision Boundaries

$$\begin{aligned} & \operatorname*{argmax}_y P(y|x) \\ = & \operatorname*{argmax}_y \exp\left(\theta_y + \sum_i \theta_{y,i} x_i\right) \end{aligned}$$

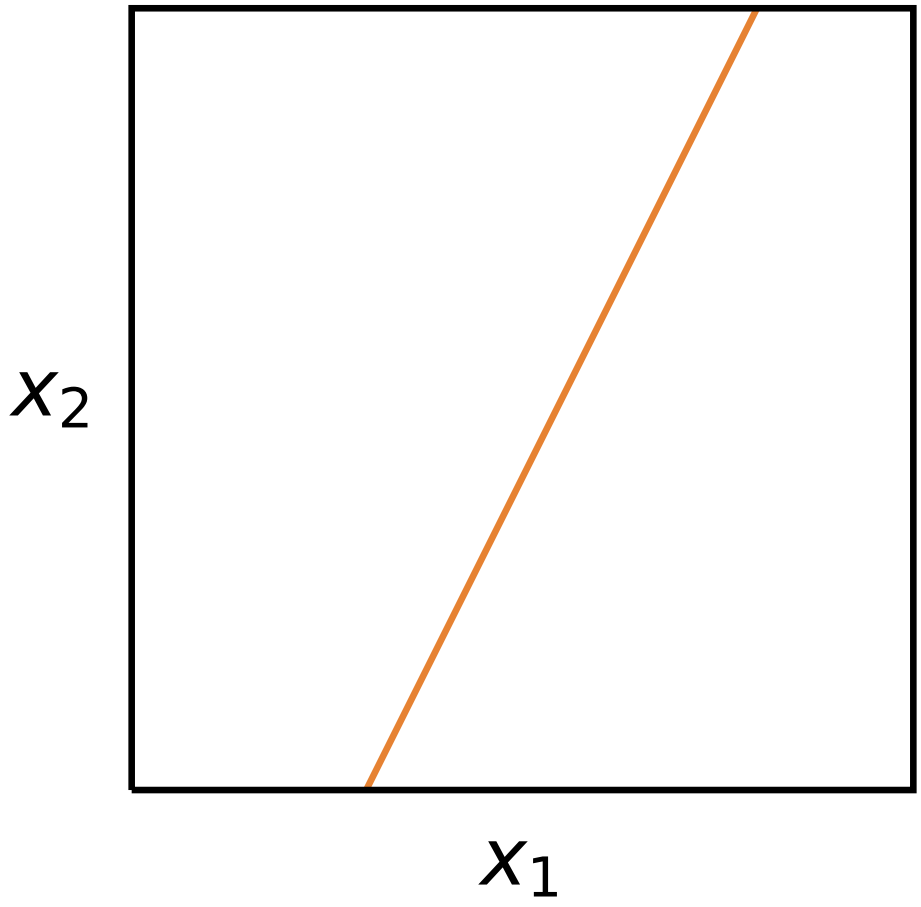
Decision Boundaries

$$\begin{aligned} & \operatorname*{argmax}_y P(y|x) \\ = & \operatorname*{argmax}_y \exp\left(\theta_y + \sum_i \theta_{y,i} x_i\right) \end{aligned}$$

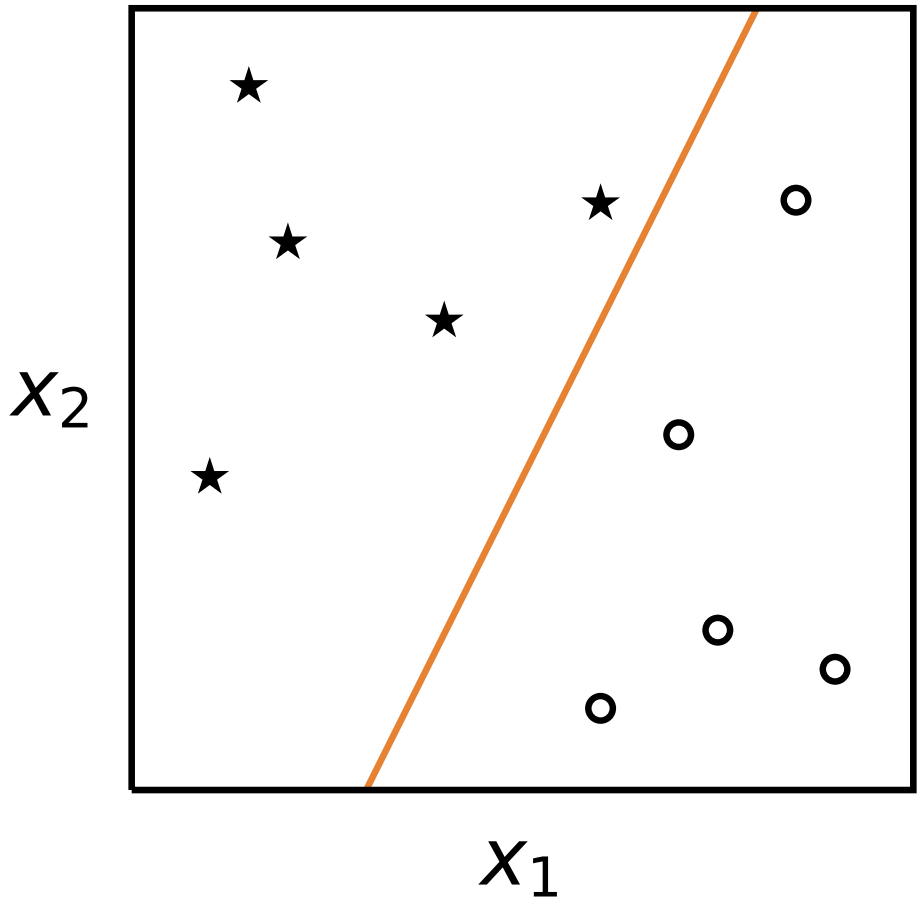
For two classes:

$$b + \sum_i a_i x_i > 0 \ ?$$

Decision Boundaries



Decision Boundaries



Support Vector Machines

- Non-probabilistic
- $\text{argmax}_y \left(\theta_y + \sum_i \theta_{y,i} x_i \right)$

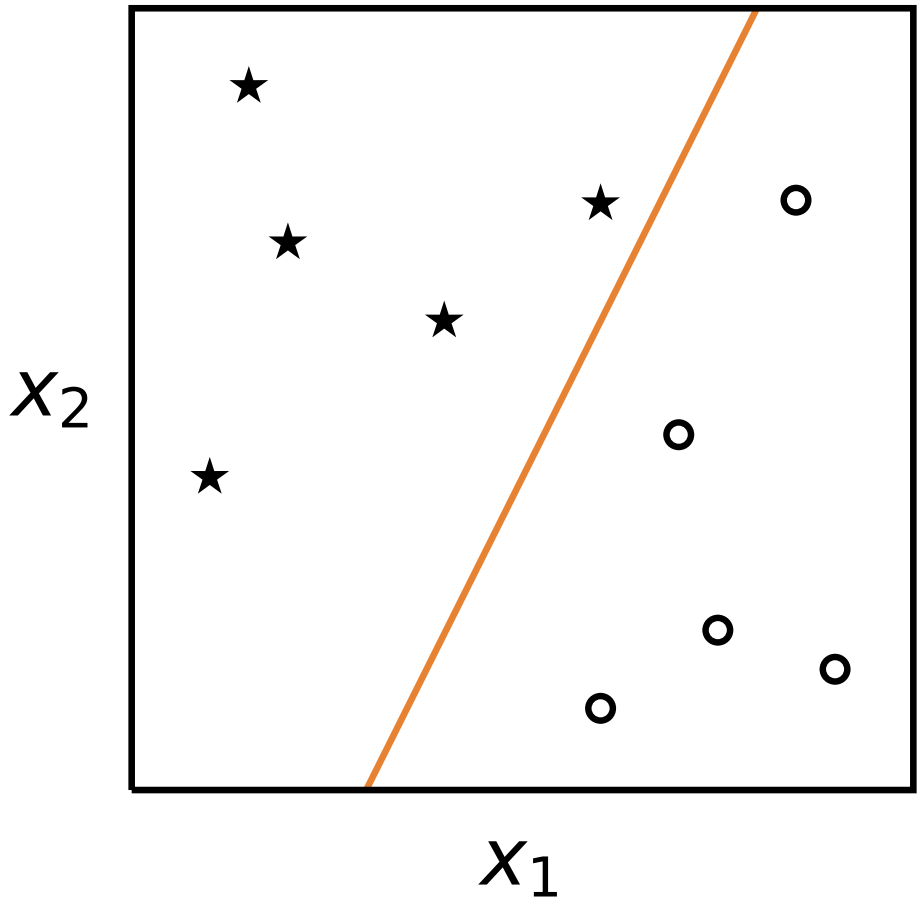
Support Vector Machines

- Non-probabilistic
- $\text{argmax}_y \left(\theta_y + \sum_i \theta_{y,i} x_i \right)$
- Linear: hyperplane boundary

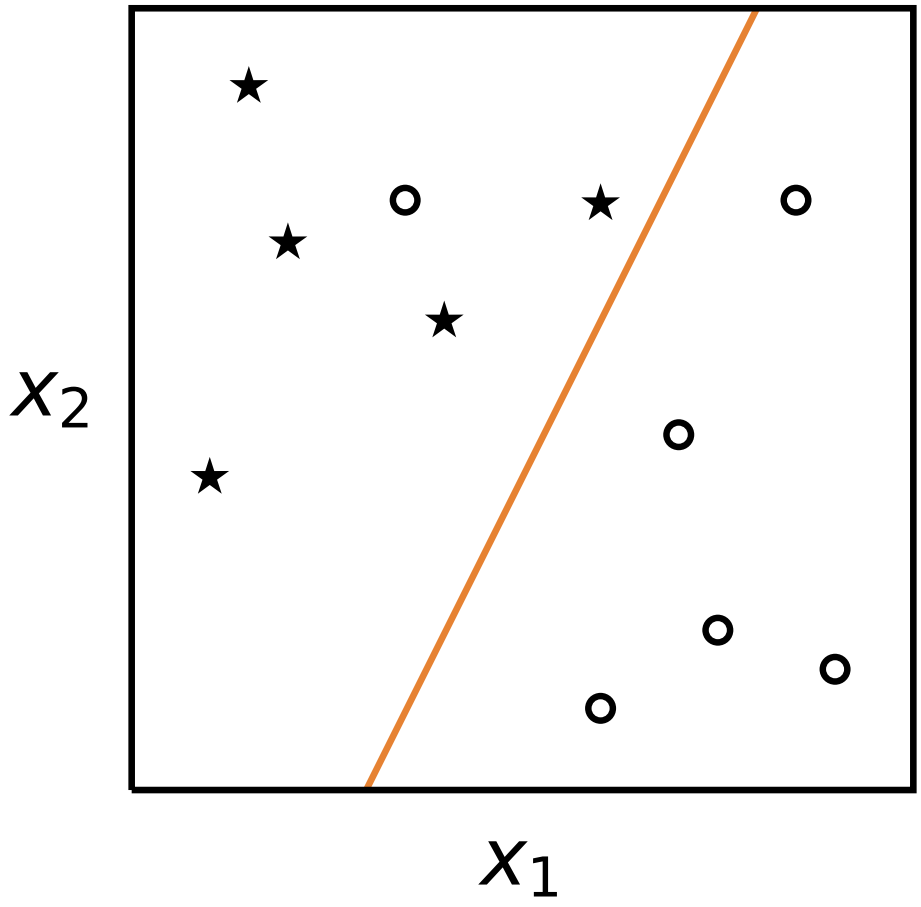
Perceptron Algorithm

- Iterate through training data
- If correct, do nothing
- If incorrect, update:
  - Correct class y: $\theta_{y,i} \leftarrow \theta_{y,i} + x_i$
  - Incorrect class y': $\theta_{y',i} \leftarrow \theta_{y',i} - x_i$

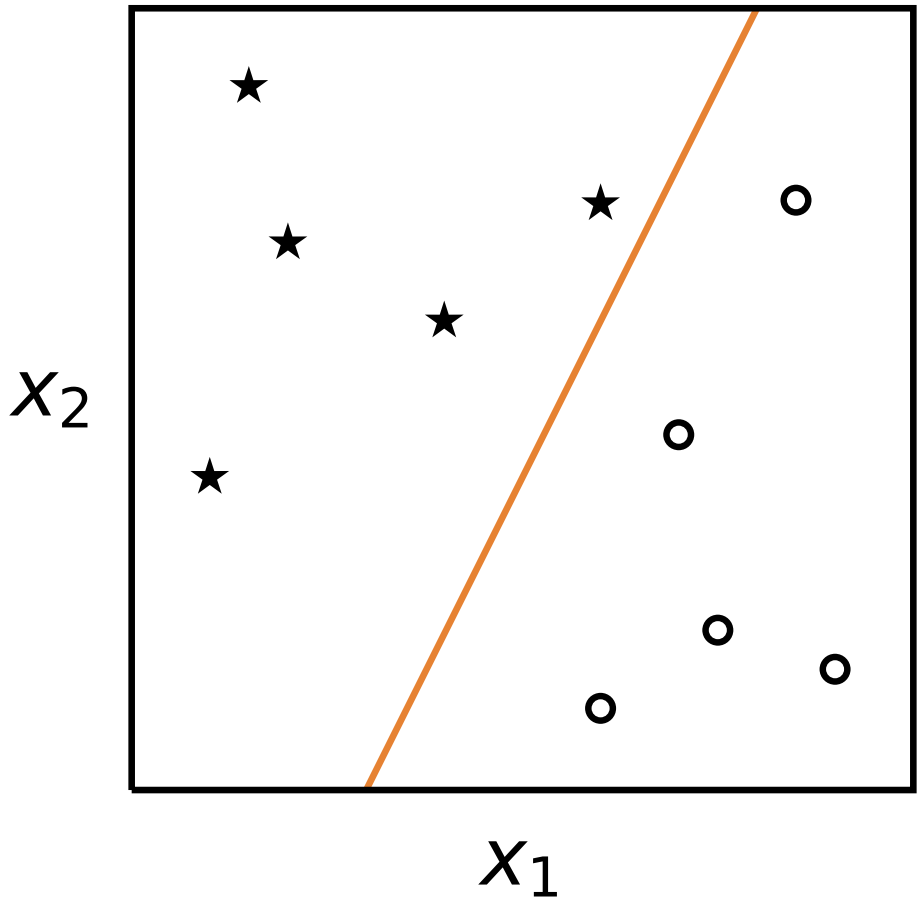
Linear Separability



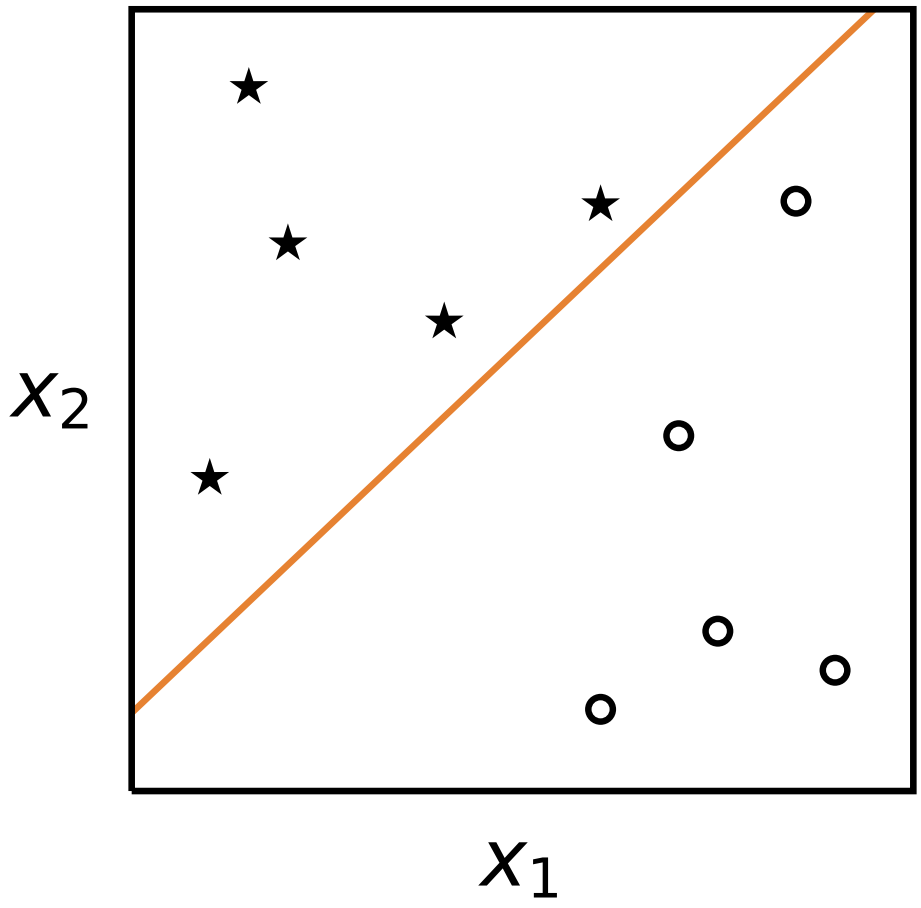
Linear Separability



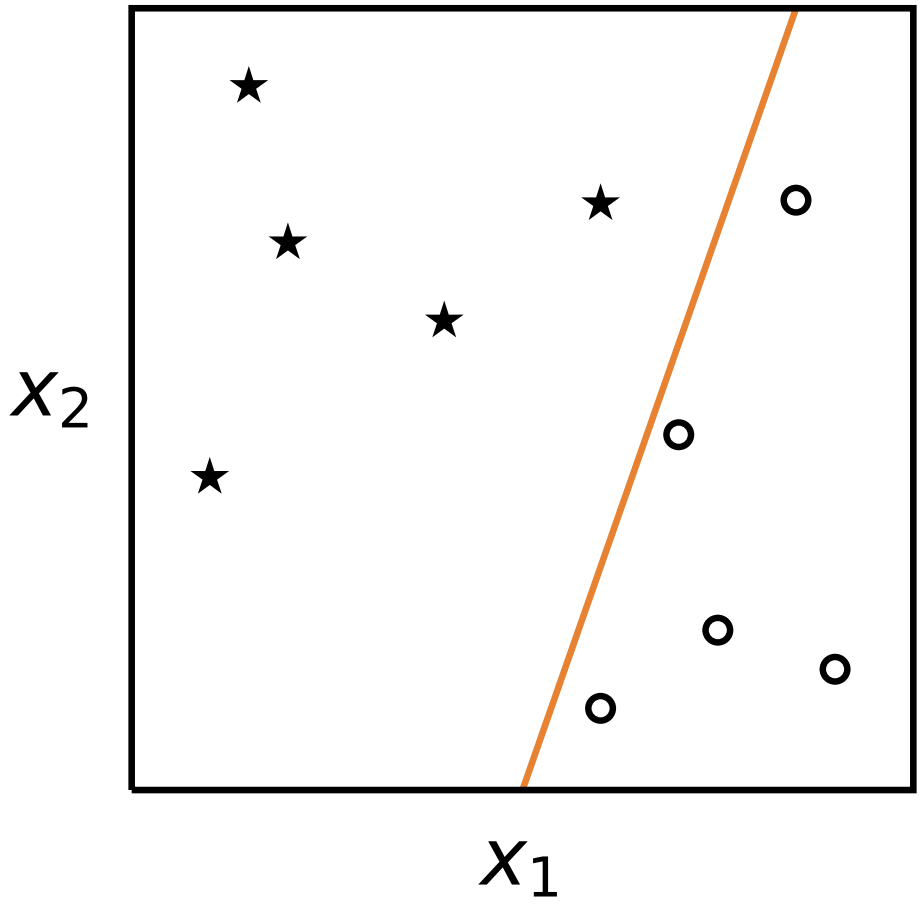
Linear Separability



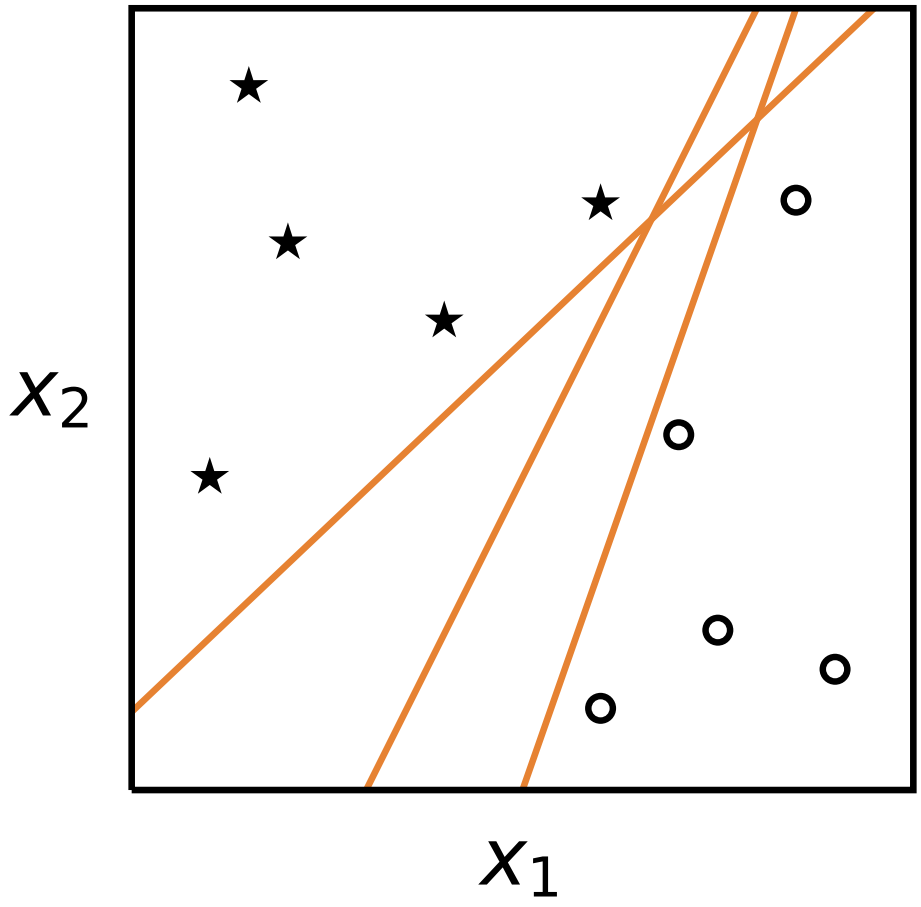
Linear Separability



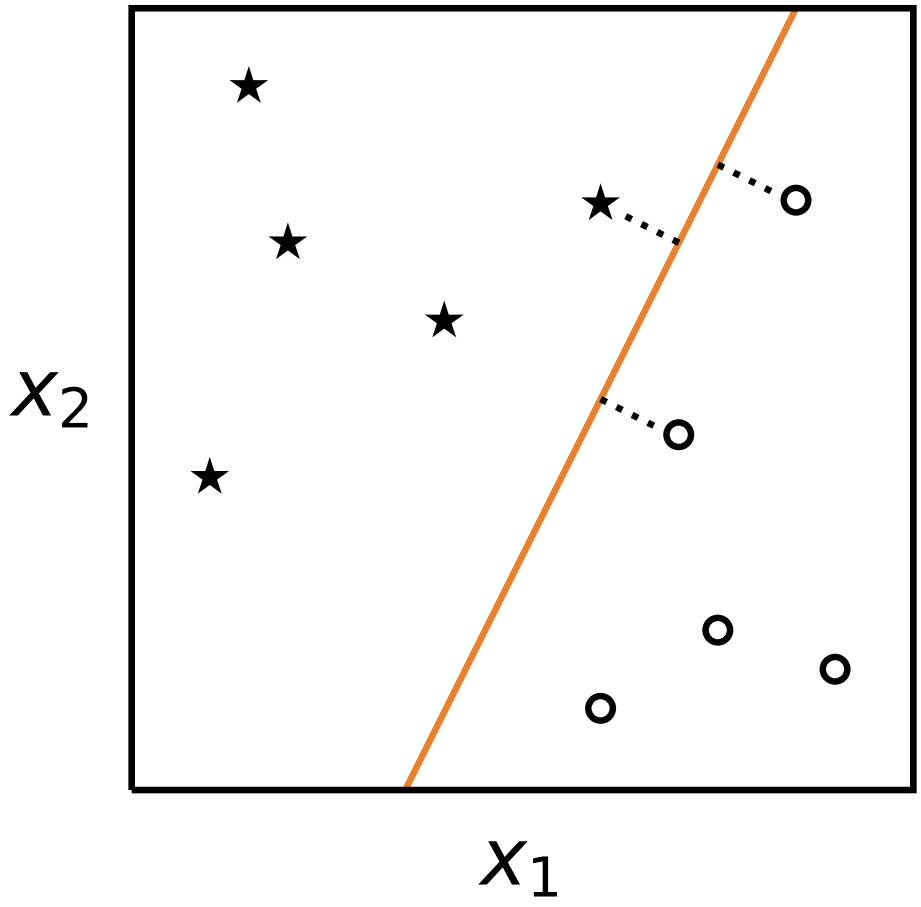
Linear Separability



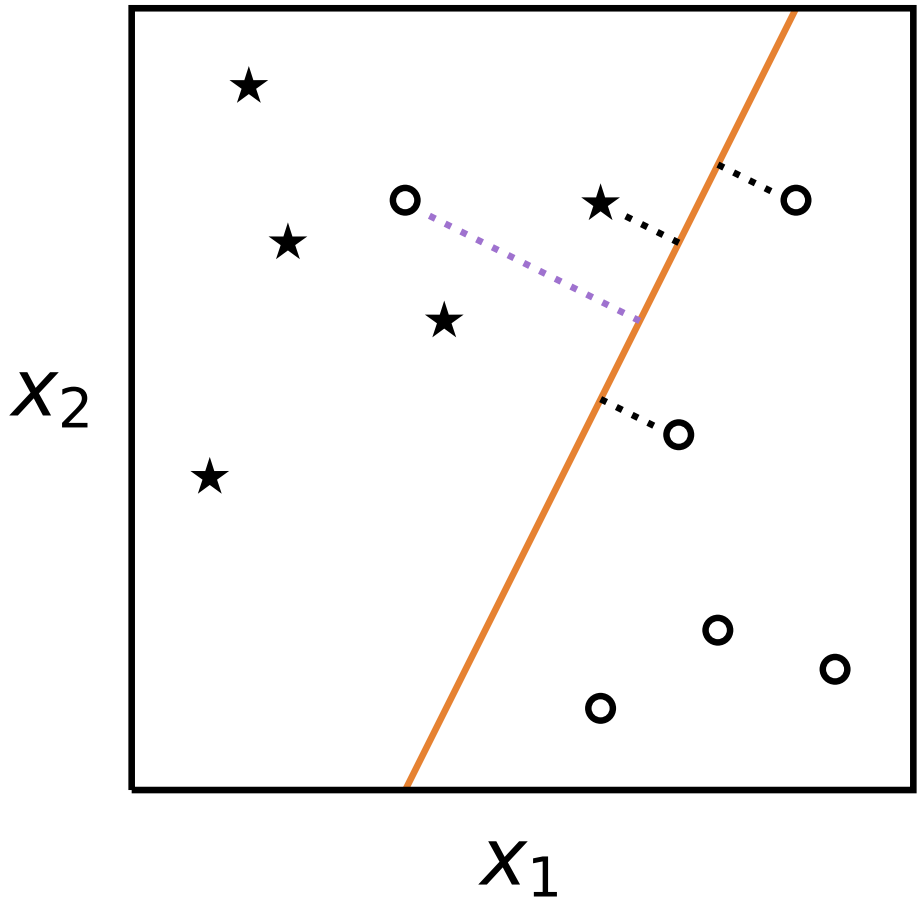
Linear Separability



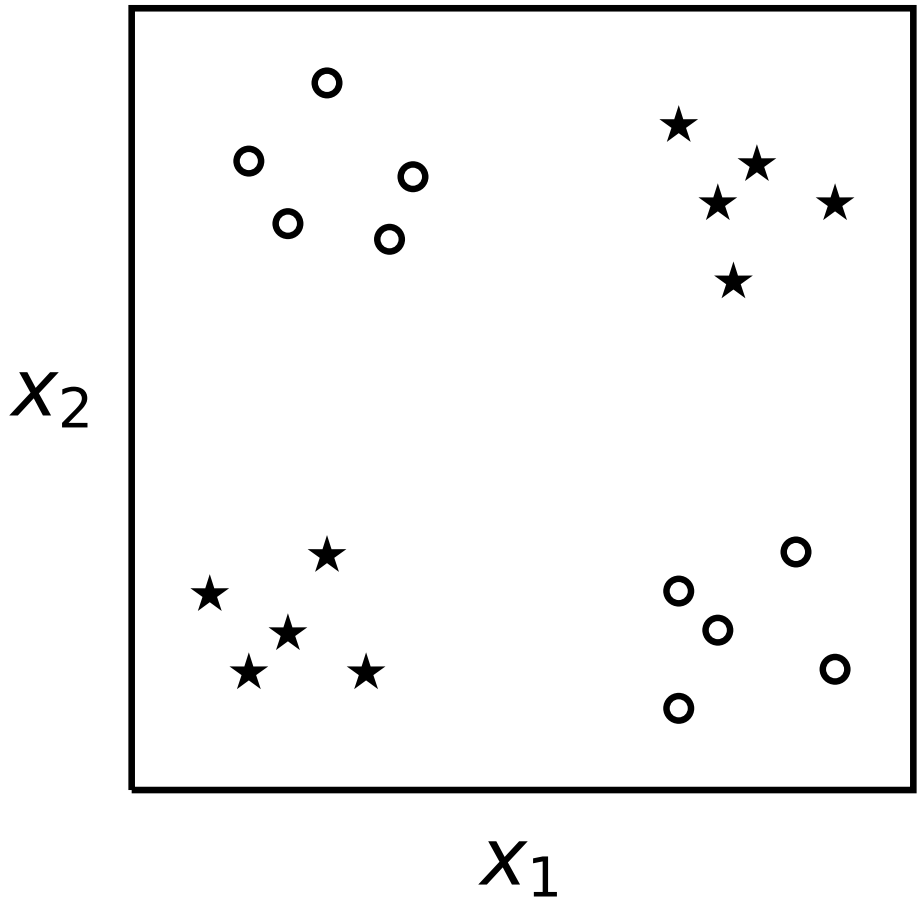
Maximum Margin



Maximum Margin



Nonlinear Decision Boundaries



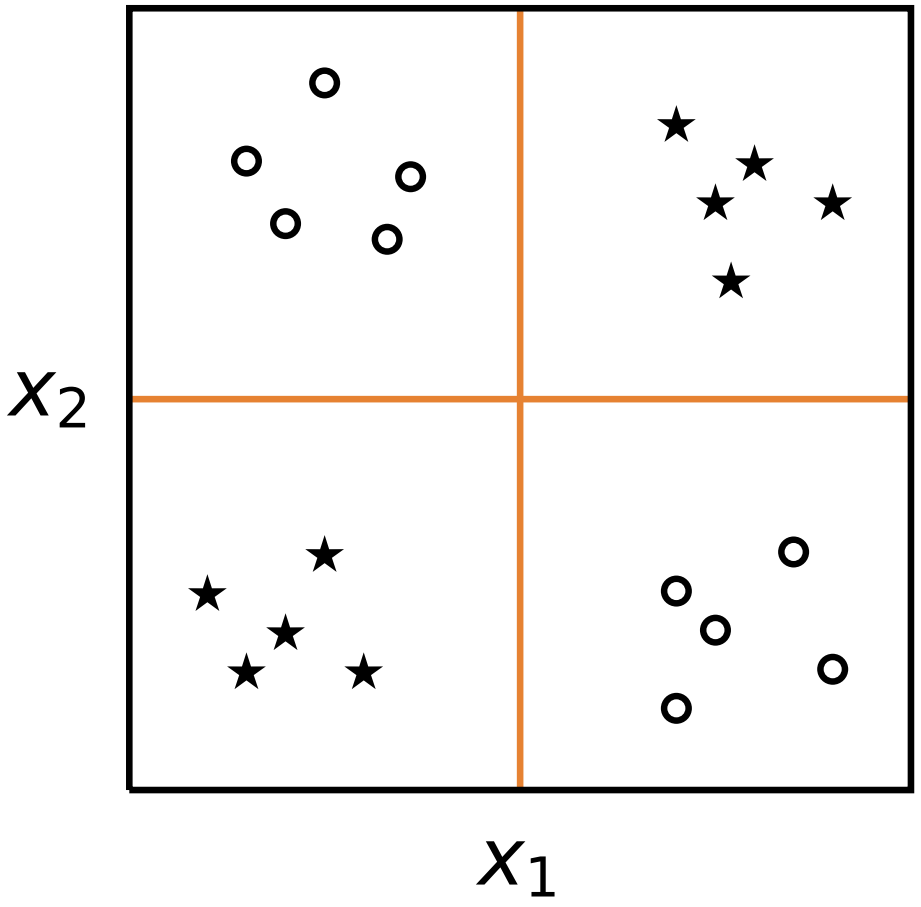
Kernel Methods

- Input space → Feature space
- Linear boundary in feature space

Kernel Methods

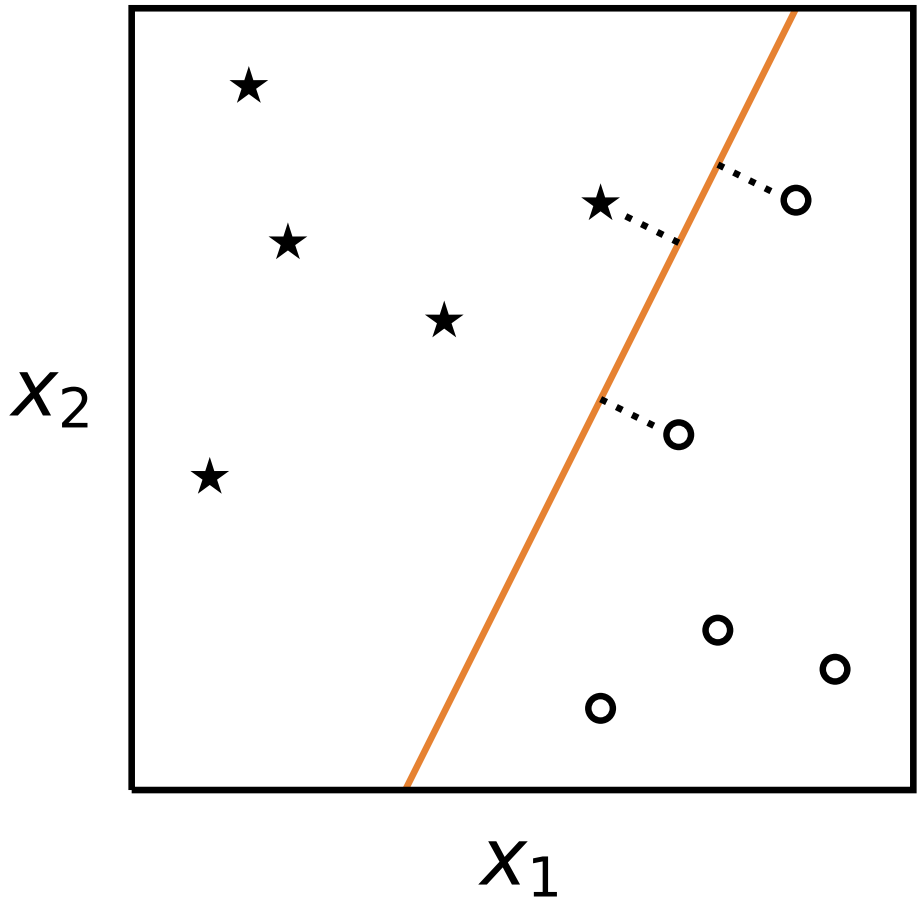
- Input space → Feature space
- Linear boundary in feature space
- e.g. $(x_1, x_2) \mapsto (x_1, x_2, x_1 x_2)$

Kernel Methods

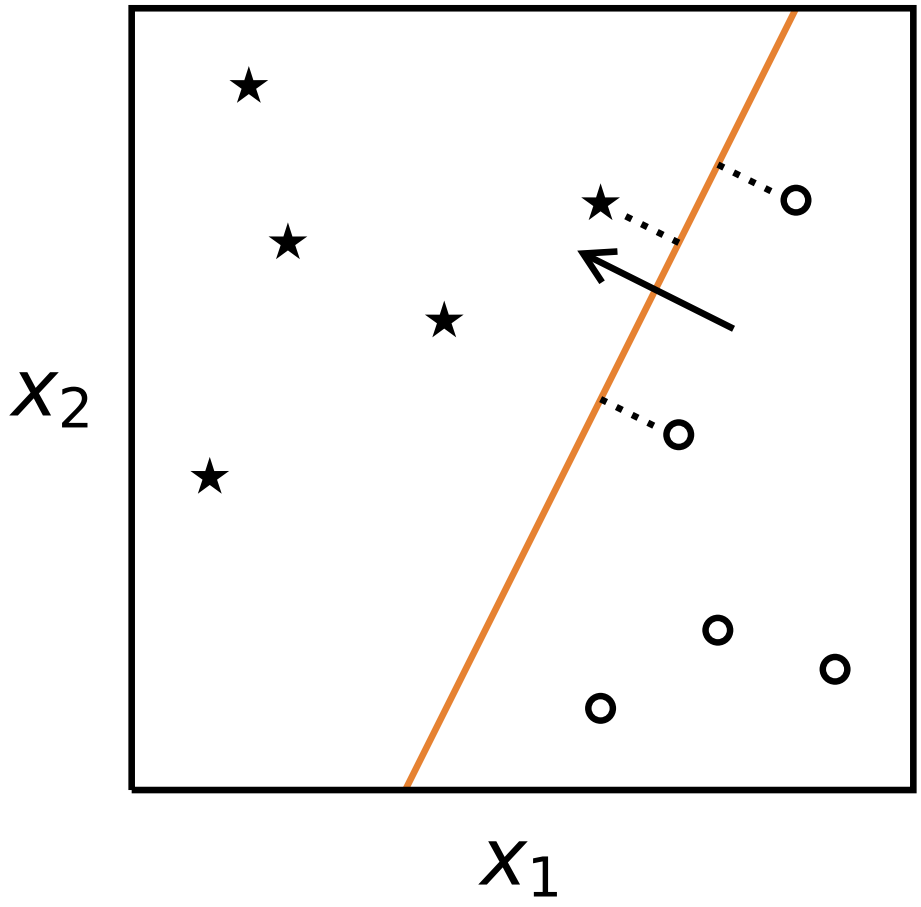


$$xy - x - y + 1 = 0$$

Maximum Margin



Maximum Margin



Kernel Methods

- Can represent and train an SVM only using dot products in feature space

Kernel Methods

- Can represent and train an SVM only using dot products in feature space
- Kernel function in input space = dot product in feature space

String Kernel

Number of shared substrings

String Kernel

Number of shared substrings

The dog barked

The dog slept

String Kernel

Number of shared substrings

The dog barked

The dog slept

String Kernel

Number of shared substrings

The dog barked

The dog slept

String Kernel

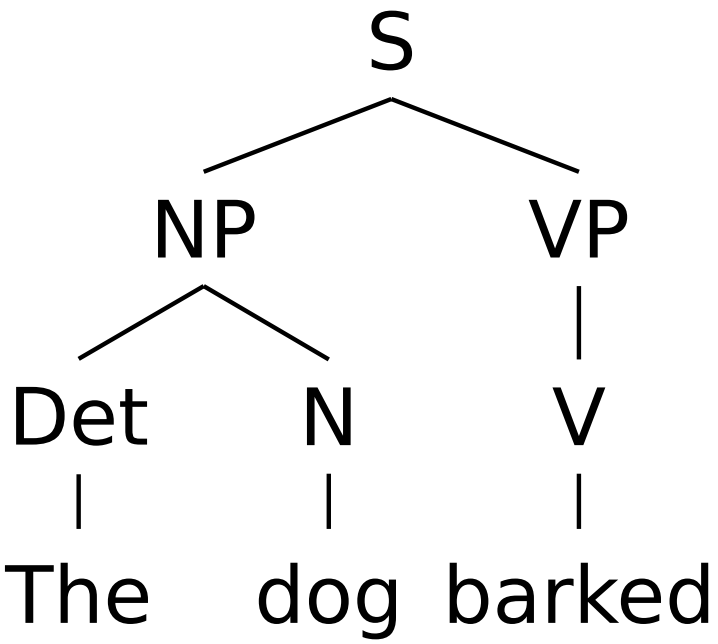
Number of shared substrings

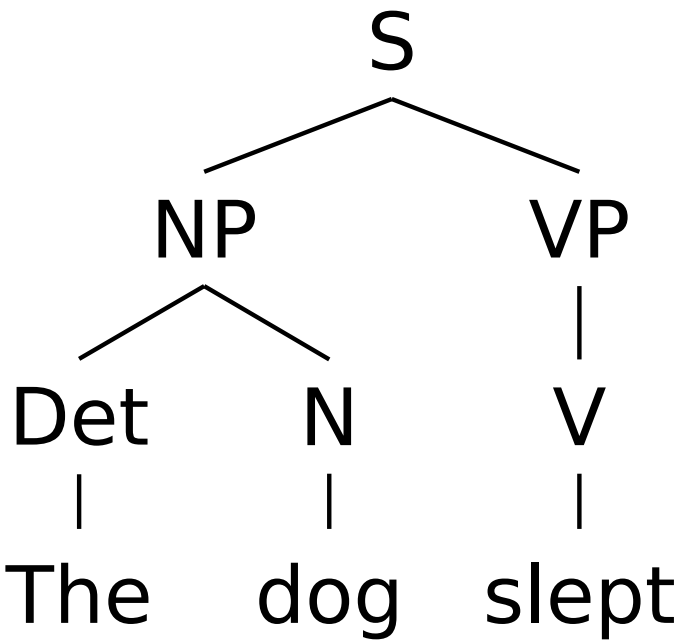
The dog barked

The dog slept

Tree Kernel

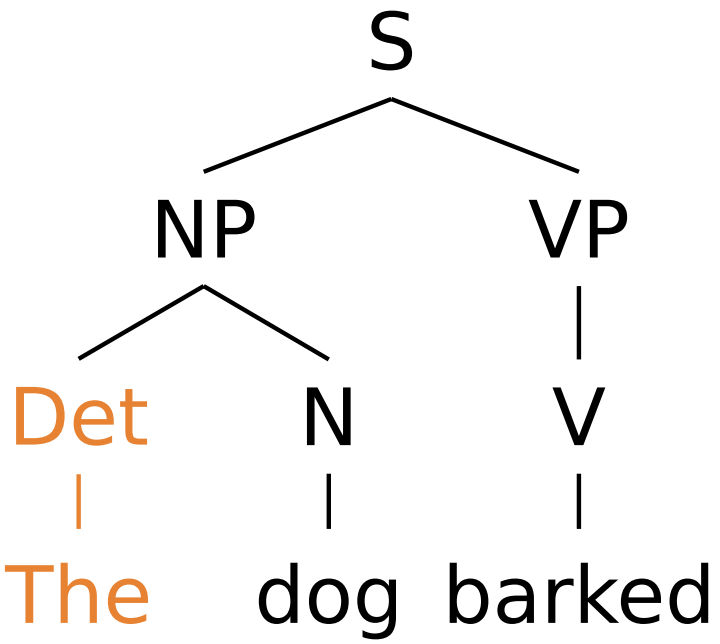
Number of shared subtrees

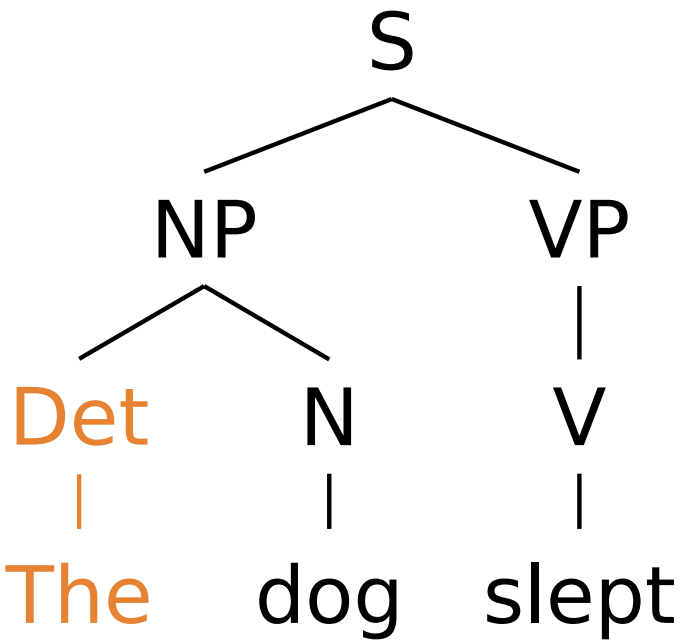




Tree Kernel

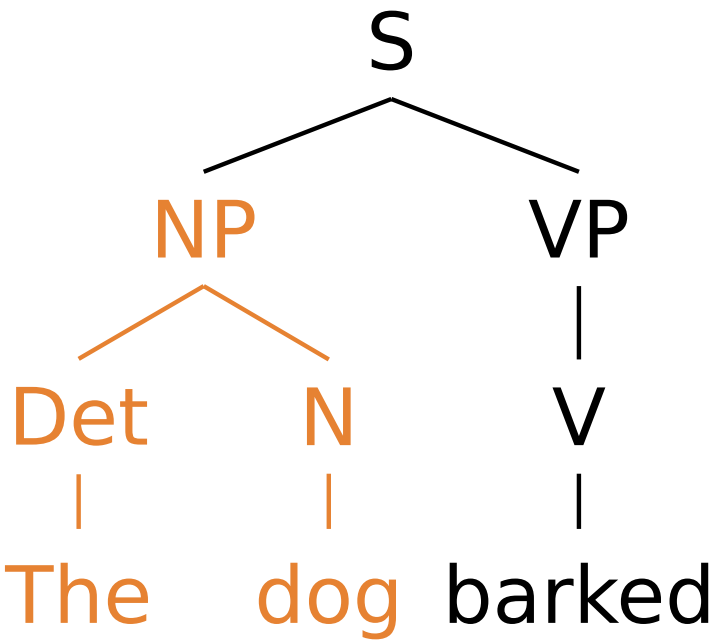
Number of shared subtrees

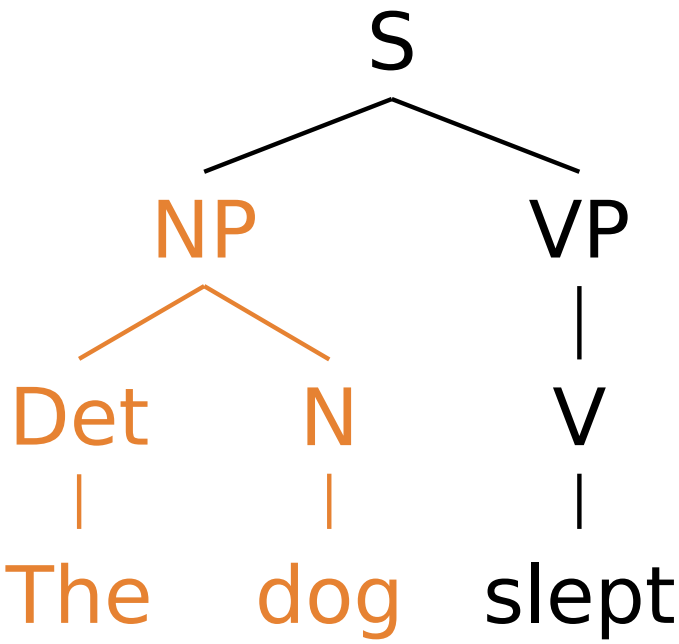




Tree Kernel

Number of shared subtrees





Tree Kernel

Number of shared subtrees