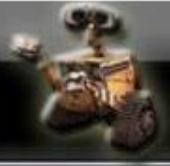


Artificial Intelligence

Combining Models

Donghui Wang
AI Institute@ZJU
2015.4

Contents

- Bayesian model averaging
- Committees
- Boosting
- Tree-based models (*Decision tree*)

References:

1. *Bishop. "Pattern Recognition and Machine Learning", Chapter 14. 2006.*
2. *Stuart J. Russell and Peter Norvig. "Artificial Intelligence: A Modern Approach", Chapter 18. 2011*

Introduction

- We have explored a range of different models for solving classification and regression problems.
- It is often found that improved performance can be obtained by combining multiple models together in some way, instead of just using a single model in isolation.
  - *We might train L different models and then make predictions using the average of the predictions made by each model. (committees)*
  - *One important variant of the committee method, known as boosting, involves training multiple models in sequence in which the error function used to train a particular model depends on the performance of the previous models. This can produce substantial improvements in performance compared to the use of a single model.*
  - *select one of the models to make the prediction, in which the choice of model is a function of the input variables. Thus different models become responsible for making predictions in different regions of input space. One widely used framework of this kind is known as a decision tree in which the selection process can be described as a sequence of binary selections corresponding to the traversal of a tree structure.*

Bayesian model averaging

Model combination vs. Bayesian model averaging

- *Model combination*: $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$

$$p(\mathbf{x}, \mathbf{z}) \Rightarrow p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}) \Rightarrow p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

  - each observed data point $\mathbf{x}_n$ has a corresponding latent variable $\mathbf{z}_n$

$$p(\mathbf{X}) = \prod_{n=1}^{N} p(\mathbf{x}_n) = \prod_{n=1}^{N} \left[\sum_{\mathbf{z}_n} p(\mathbf{x}_n, \mathbf{z}_n) \right]$$

- *Bayesian model averaging*:
  - Suppose we have several different models indexed by $h = 1, \ldots, H$ with prior probabilities $p(h)$, then

$$p(\mathbf{X}) = \sum_{h=1}^{H} p(\mathbf{X}|h)p(h)$$

Committees

Bootstrap aggregation / Bagging

- Bootstrap data set:
  - Suppose our original data set consists of N data points $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. We can create a new data set $\mathbf{X}_B$ by drawing N points at random from $\mathbf{X}$, with replacement, so that some points in $\mathbf{X}$ may be replicated in $\mathbf{X}_B$, whereas other points in $\mathbf{X}$ may be absent from $\mathbf{X}_B$. This process can be repeated L times to generate L data sets each of size N and each obtained by sampling from the original data set $\mathbf{X}$.
- Committee prediction:

$$y_{\mathrm{COM}}(\mathbf{x}) = \frac{1}{M}\sum_{m=1}^{M} y_m(\mathbf{x})$$

- Suppose the true regression function h(x) and the output of each of the models is:

$$y_m(\mathbf{x}) = h(\mathbf{x}) + \epsilon_m(\mathbf{x})$$

- The average SSE:

$$\mathbb{E}_{\mathbf{x}}\left[\{y_m(\mathbf{x}) - h(\mathbf{x})\}^2\right] = \mathbb{E}_{\mathbf{x}}\left[\epsilon_m(\mathbf{x})^2\right] \quad \Rightarrow \quad E_{\mathrm{AV}} = \frac{1}{M}\sum_{m=1}^{M} \mathbb{E}_{\mathbf{x}}\left[\epsilon_m(\mathbf{x})^2\right]$$

Bootstrap aggregation / Bagging

- Committee prediction: $$y_{\mathrm{COM}}(\mathbf{x}) = \frac{1}{M}\sum_{m=1}^{M} y_m(\mathbf{x})$$

- The average SSE: $y_m(\mathbf{x}) = h(\mathbf{x}) + \epsilon_m(\mathbf{x})$

$$\mathbb{E}_{\mathbf{x}}\left[\{y_m(\mathbf{x}) - h(\mathbf{x})\}^2\right] = \mathbb{E}_{\mathbf{x}}\left[\epsilon_m(\mathbf{x})^2\right] \quad \Rightarrow \quad E_{\mathrm{AV}} = \frac{1}{M}\sum_{m=1}^{M}\mathbb{E}_{\mathbf{x}}\left[\epsilon_m(\mathbf{x})^2\right]$$

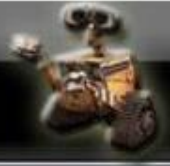
$$E_{\mathrm{COM}} = \mathbb{E}_{\mathbf{x}}\left[\left\{\frac{1}{M}\sum_{m=1}^{M} y_m(\mathbf{x}) - h(\mathbf{x})\right\}^2\right] = \mathbb{E}_{\mathbf{x}}\left[\left\{\frac{1}{M}\sum_{m=1}^{M}\epsilon_m(\mathbf{x})\right\}^2\right]$$

- Assume the errors have zero mean and are *uncorrelated*:

$$\mathbb{E}_{\mathbf{x}}\left[\epsilon_m(\mathbf{x})\right] = 0$$
$$\mathbb{E}_{\mathbf{x}}\left[\epsilon_m(\mathbf{x})\epsilon_l(\mathbf{x})\right] = 0,\ m \neq l$$

$$\Rightarrow \quad \boxed{E_{\mathrm{COM}} = \frac{1}{M}E_{\mathrm{AV}}}$$

Boosting



Boosting

- Boosting is a powerful technique for combining multiple 'base' classifiers to produce a form of committee whose performance can be significantly better than that of any of the base classifiers.
  - *AdaBoost*: adaptive boosting
  - E.g. Combining many weak classifiers to form a strong classifier

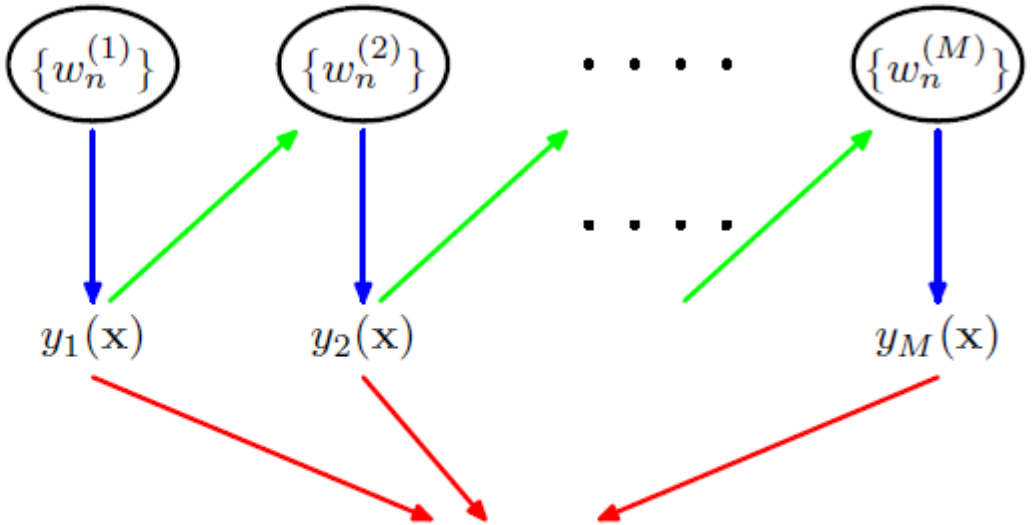
---

- Boosting vs. Committee
  - The base classifiers are trained in sequence
  - Each base classifier is trained using a weighed form of the data set in which the weighing coefficient associated with each data point depends on the performance of the previous classifiers
    - Misclassified points get greater weight in next training

Boosting

- Combining the predictions of all classifiers through a weighted majority voting scheme:

Schematic illustration of the boosting framework. Each base classifier $y_m(\mathbf{x})$ is trained on a weighted form of the training set (blue arrows) in which the weights $w_n^{(m)}$ depend on the performance of the previous base classifier $y_{m-1}(\mathbf{x})$ (green arrows). Once all base classifiers have been trained, they are combined to give the final classifier $Y_M(\mathbf{x})$ (red arrows).



$$Y_M(\mathbf{x}) = \text{sign}\left(\sum_m^M \alpha_m y_m(\mathbf{x})\right)$$

Boosting

- Algorithm:

1. Initialize the data weighting coefficients $\{w_n\}$ by setting $w_n^{(1)} = 1/N$ for $n = 1, \ldots, N$.

2. For $m = 1, \ldots, M$:

   (a) Fit a classifier $y_m(\mathbf{x})$ to the training data by minimizing the weighted error function

   $$J_m = \sum_{n=1}^{N} w_n^{(m)} I(y_m(\mathbf{x}_n) \neq t_n)$$

   where $I(y_m(\mathbf{x}_n) \neq t_n)$ is the indicator function and equals 1 when $y_m(\mathbf{x}_n) \neq t_n$ and 0 otherwise.

   (b) Evaluate the quantities

   $$\epsilon_m = \sum_{n=1}^{N} w_n^{(m)} I(y_m(\mathbf{x}_n) \neq t_n) \Big/ \sum_{n=1}^{N} w_n^{(m)}$$

   and then use these to evaluate

   $$\alpha_m = \ln \left\{ \frac{1 - \epsilon_m}{\epsilon_m} \right\}.$$

   (c) Update the data weighting coefficients

   $$w_n^{(m+1)} = w_n^{(m)} \exp \{\alpha_m I(y_m(\mathbf{x}_n) \neq t_n)\}$$

3. Make predictions using the final model, which is given by

   $$Y_M(\mathbf{x}) = \text{sign} \left(\sum_{m=1}^{M} \alpha_m y_m(\mathbf{x}) \right).$$

Boosting

- Example results:

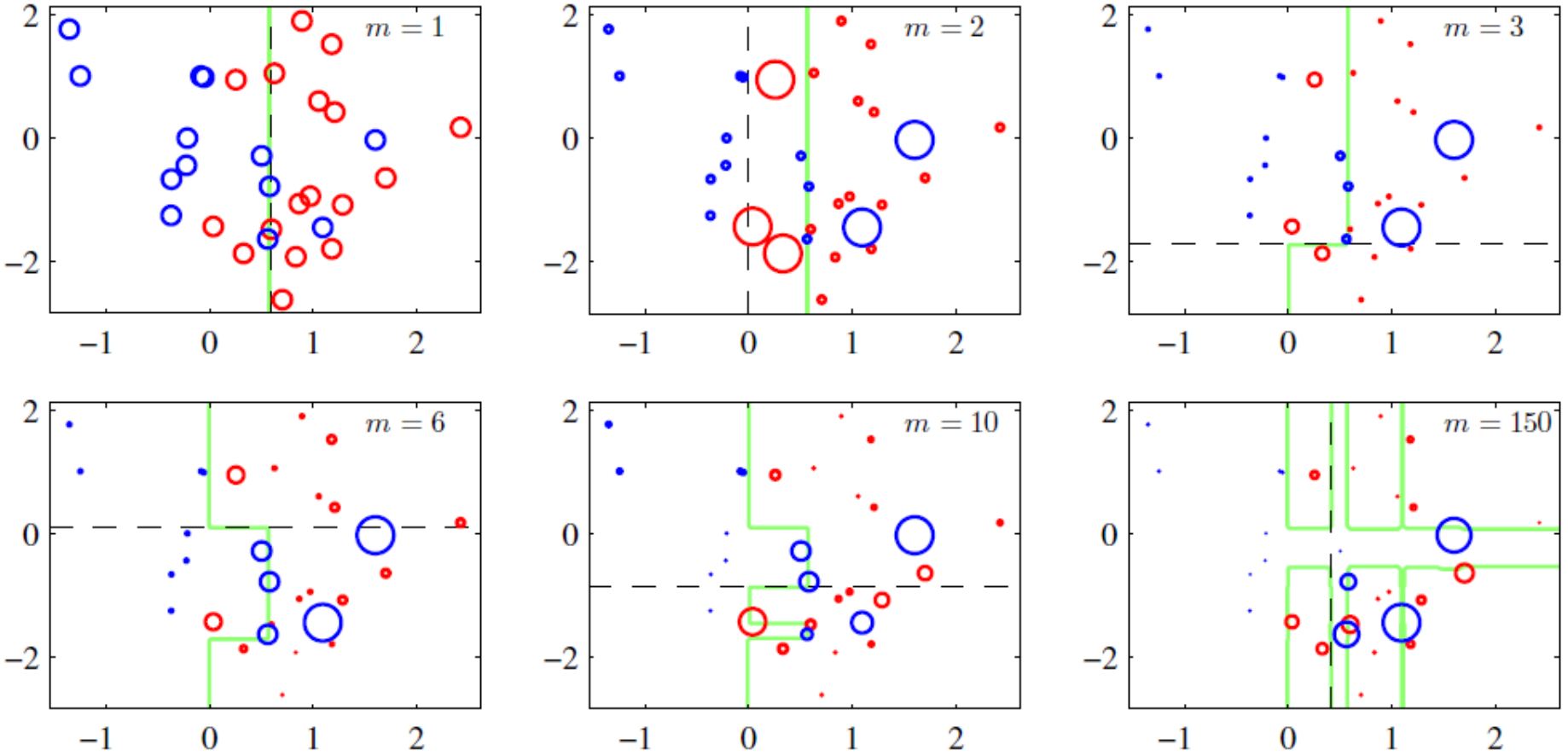


Figure 14.2 Illustration of boosting in which the base learners consist of simple thresholds applied to one or other of the axes. Each figure shows the number m of base learners trained so far, along with the decision boundary of the most recent base learner (dashed black line) and the combined decision boundary of the ensemble (solid green line). Each data point is depicted by a circle whose radius indicates the weight assigned to that data point when training the most recently added base learner. Thus, for instance, we see that points that are misclassified by the $m = 1$ base learner are given greater weight when training the $m = 2$ base learner.

Minimizing exponential error

- Consider the exponential error function defined by:

Base classifier

$$E = \sum_{n=1}^{N} \exp\{-t_n f_m(\mathbf{x}_n)\} \qquad f_m(\mathbf{x}) = \frac{1}{2}\sum_{l=1}^{m} \alpha_l y_l(\mathbf{x}) \qquad t_n \in \{-1, 1\}$$

- Goal: minimize E with respect to both the weighting coefficients α_l and the parameters of the base classifiers $y_l(\mathbf{x})$.

- Minimizing only with respect to the m^{th} base classifier:

$\exp\{-t_n f_{m-1}(\mathbf{x}_n)\}$

$$E = \sum_{n=1}^{N} \exp\left\{-t_n f_{m-1}(\mathbf{x}_n) - \frac{1}{2} t_n \alpha_m y_m(\mathbf{x}_n)\right\} = \sum_{n=1}^{N} w_n^{(m)} \exp\left\{-\frac{1}{2} t_n \alpha_m y_m(\mathbf{x}_n)\right\}$$

$$\Rightarrow \quad E = e^{-\alpha_m/2} \sum_{n \in \mathcal{T}_m} w_n^{(m)} + e^{\alpha_m/2} \sum_{n \in \mathcal{M}_m} w_n^{(m)}$$

$$= (e^{\alpha_m/2} - e^{-\alpha_m/2}) \sum_{n=1}^{N} w_n^{(m)} I(y_m(\mathbf{x}_n) \neq t_n) + e^{-\alpha_m/2} \sum_{n=1}^{N} w_n^{(m)}$$

$$w_n^{(m+1)} = w_n^{(m)} \exp\left\{-\frac{1}{2} t_n \alpha_m y_m(\mathbf{x}_n)\right\}$$

$$t_n y_m(\mathbf{x}_n) = 1 - 2I(y_m(\mathbf{x}_n) \neq t_n)$$

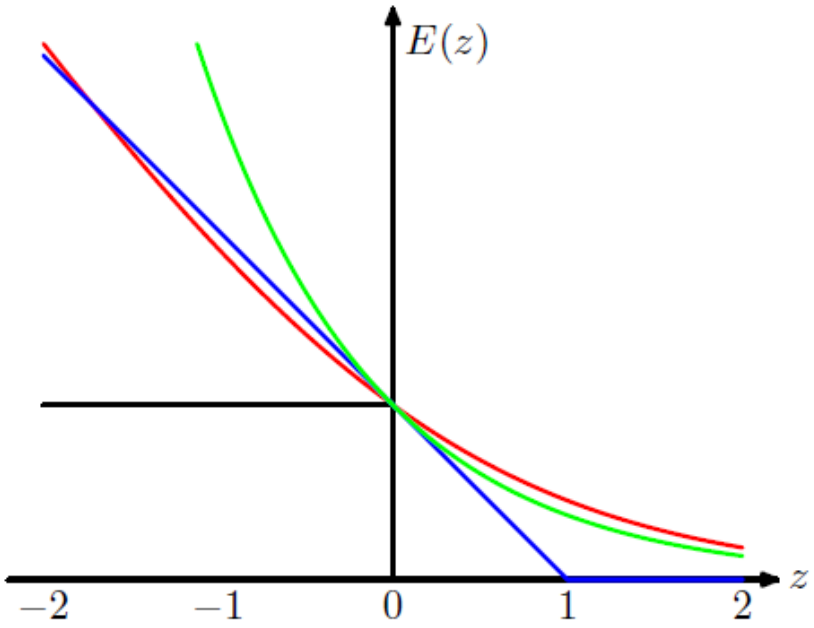
$$\Rightarrow \quad w_n^{(m+1)} = w_n^{(m)} \exp(-\alpha_m/2) \exp\{\alpha_m I(y_m(\mathbf{x}_n) \neq t_n)\}$$

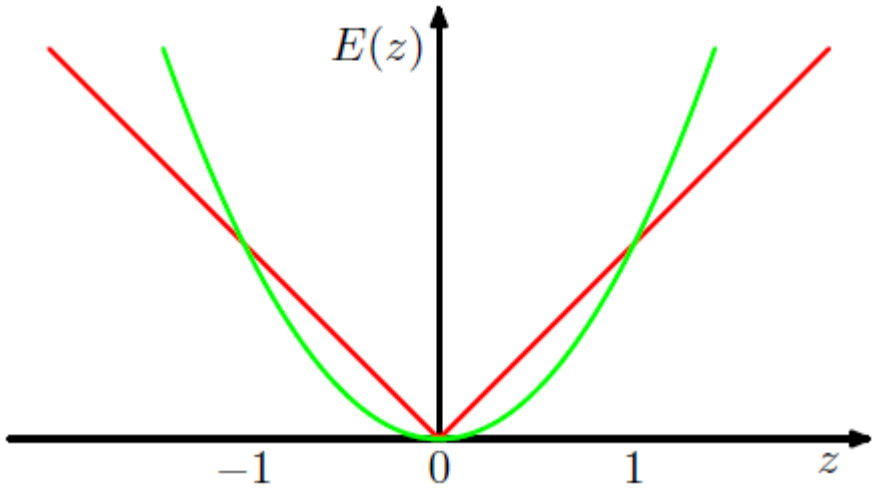
Error function for boosting

- Expected error:

$$\mathbb{E}_{\mathbf{x},t}\left[\exp\{-ty(\mathbf{x})\}\right] = \sum_t \int \exp\{-ty(\mathbf{x})\}p(t|\mathbf{x})p(\mathbf{x})\,\mathrm{d}\mathbf{x}$$

$$y(\mathbf{x}) = \frac{1}{2}\ln\left\{\frac{p(t=1|\mathbf{x})}{p(t=-1|\mathbf{x})}\right\}$$

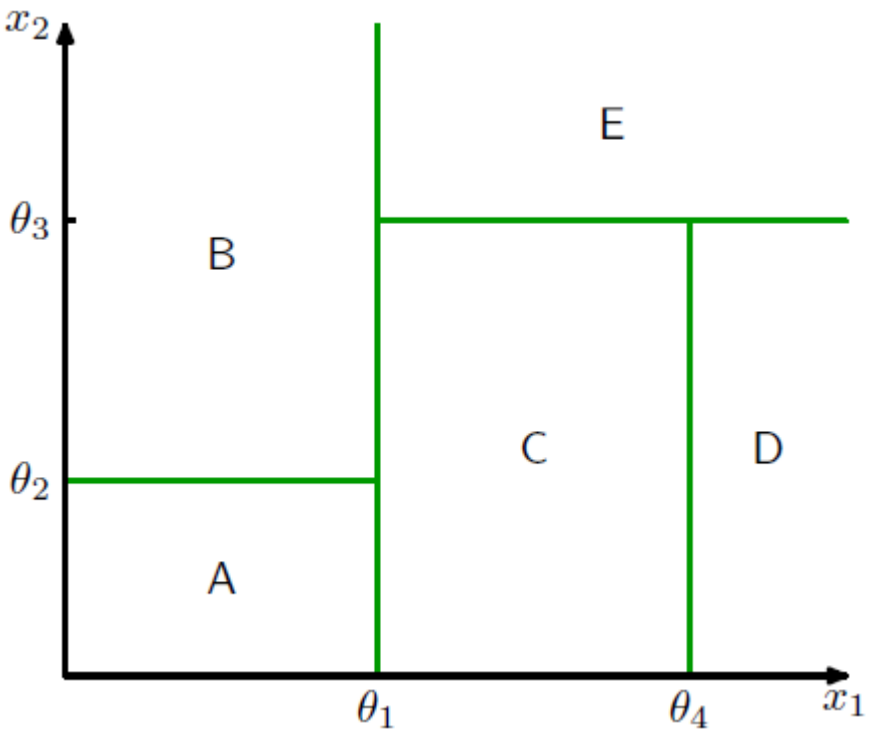


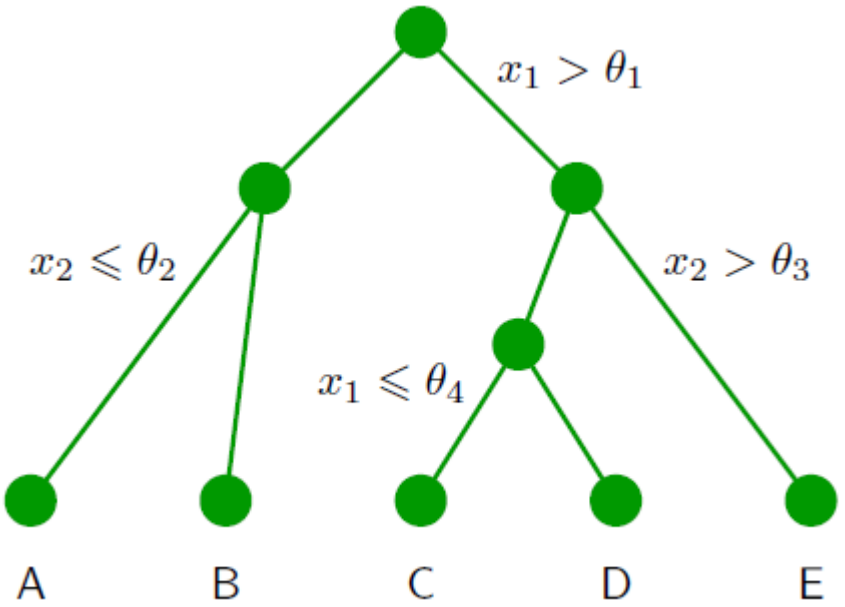


Tree-based models (*Decision tree*)

Tree-based models

- Classification and regression trees (CART):
  - Other variants: ID3, C4.5





Next: Traditional AI