Thursday, 10/22/2020

EE 660

MACHINE LEARNING FROM SIGNALS: FOUNDATIONS AND METHODS

Prof. B. Keith Jenkins

Lecture 18

p.1

Announcements

- Homework 6 (project proposal) is due tomorrow
  - Turn in a project proposal form and one dataset form for each dataset you will use
  - 1 set of documents per team (see instructions)
- Homework 7 will be posted

Today's topics

- Classification and Regression Trees (part 2) - more rigorous treatment
  - General algorithm
  - For regression (cost function)
  - For classification (cost function(s))
  - Examples
  - Tree depth, overfitting, and pruning

CART – more rigorous treatment [still Murphy 16.2]

At each iteration (each node of tree), divide one region R_m into two, by thresholding one feature x_j. Thus:

At $k^{\underline{th}}$ iteration:

$$\min_{m,\, j,\, t_k,\, w_{m_1},\, w_{m_2}} \left\{ f_{obj}^{(k)} \left(w_{m_1}, w_{m_2}, \mathcal{D}; j, t_k, m \right) \right\}$$

in which $f_{obj}^{(k)} = \text{cost}_k \left\{ (\underline{x}_i, y_i) \in \mathcal{D} \right\}_{\text{after split of } R_m}$

For cost fcns. that are additive by region, that is:

$$\text{cost}\left\{ (\underline{x}_i, y_i) \in \mathcal{D} \right\} = \alpha \sum_{m=1}^{M} \text{cost}\left\{ (\underline{x}_i, y_i) \in R_m \right\}$$

we can instead use the incremental change in cost:

$$\tilde{f}_{obj}^{(k)} = \text{cost}\left\{ (\underline{x}_i, y_i) \in R_{m_1} \right\} + \text{cost}\left\{ (\underline{x}_i, y_i) \in R_{m_2} \right\} - \text{cost}\left\{ (\underline{x}_i, y_i) \in R_m \right\}$$

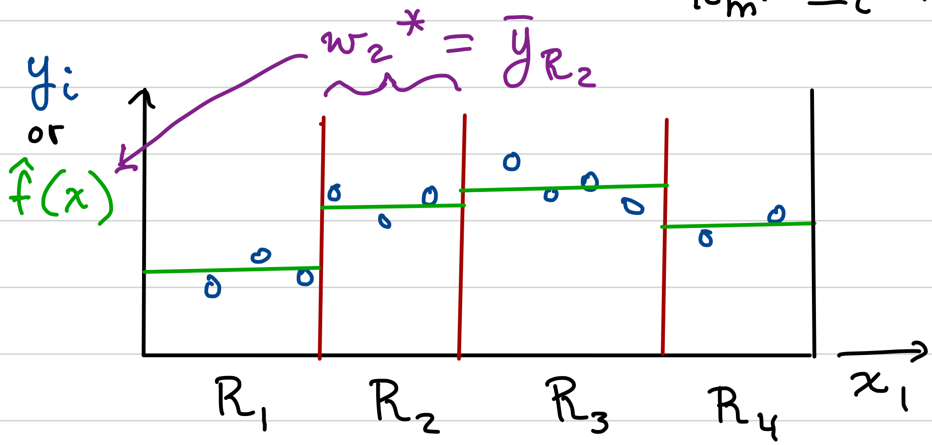
in which $R_{m_1} = R_m \cap \{ x_j \le t_k \}$; $R_{m_2} = R_m \cap \{ x_j > t_k \}$

<u>For regression</u>, cost function is typically

$$\text{cost}\{(\underline{x}_i, y_i) \in R_{m'}\} = \sum_{\underline{x}_i \in R_{m'}} (y_i - w_{m'})^2 \qquad (\text{SSE})$$

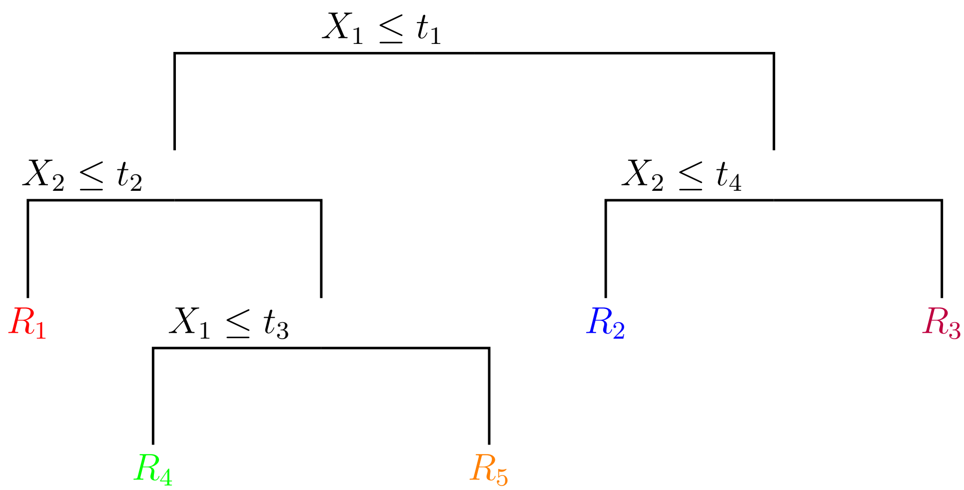
For given $R_{m'}$, this is minimized w.r.t. $w_{m'}$ by:

$$w_{m'} = w_{m'}^* = \bar{y}_{R_{m'}} \triangleq \frac{1}{N_{R_{m'}}} \sum_{\underline{x}_i \in R_{m'}} y_i$$

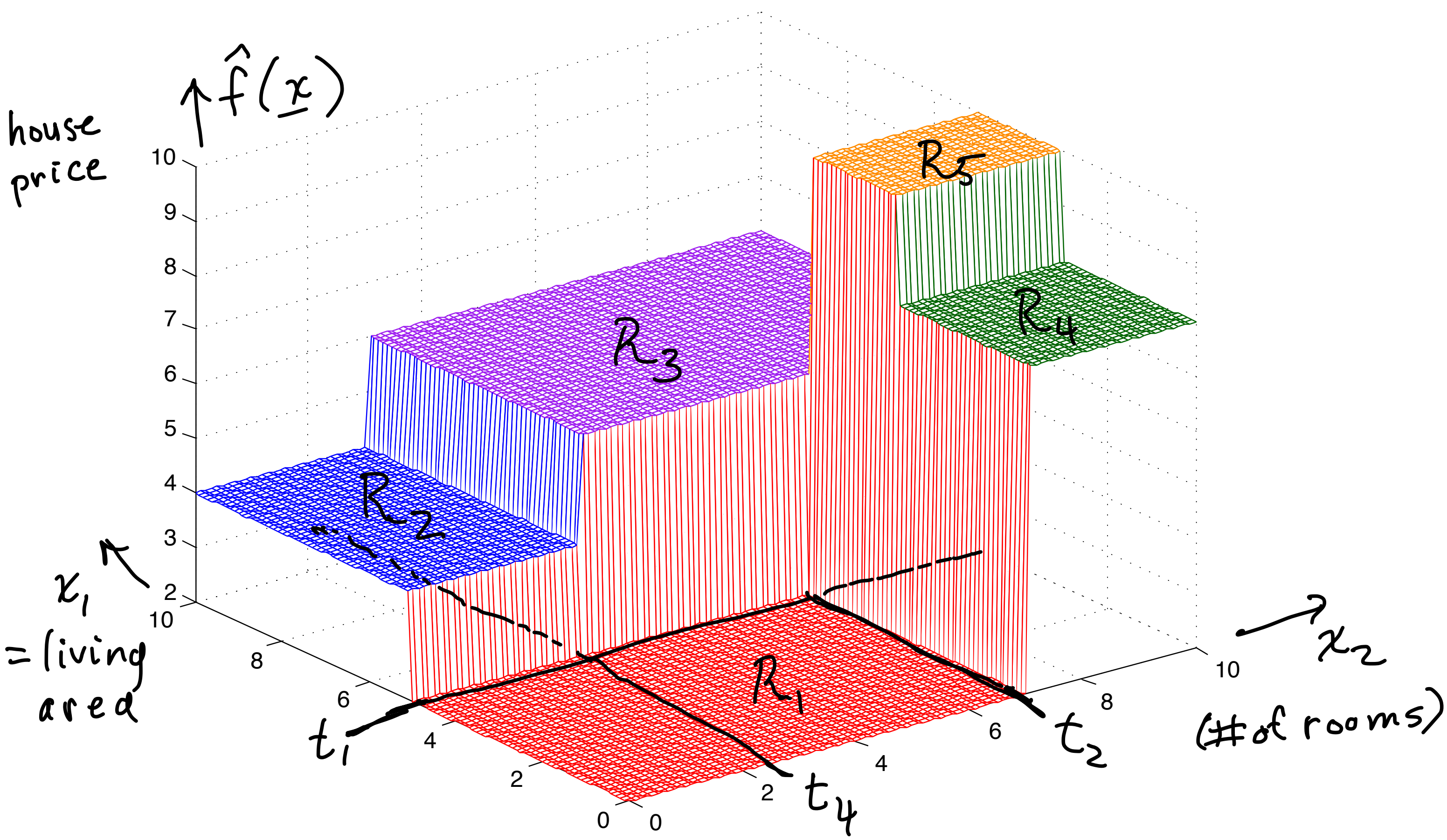


Ex: [Murphy Fig. 16.1]

Regression example



Murphy Fig. 16.1(a)



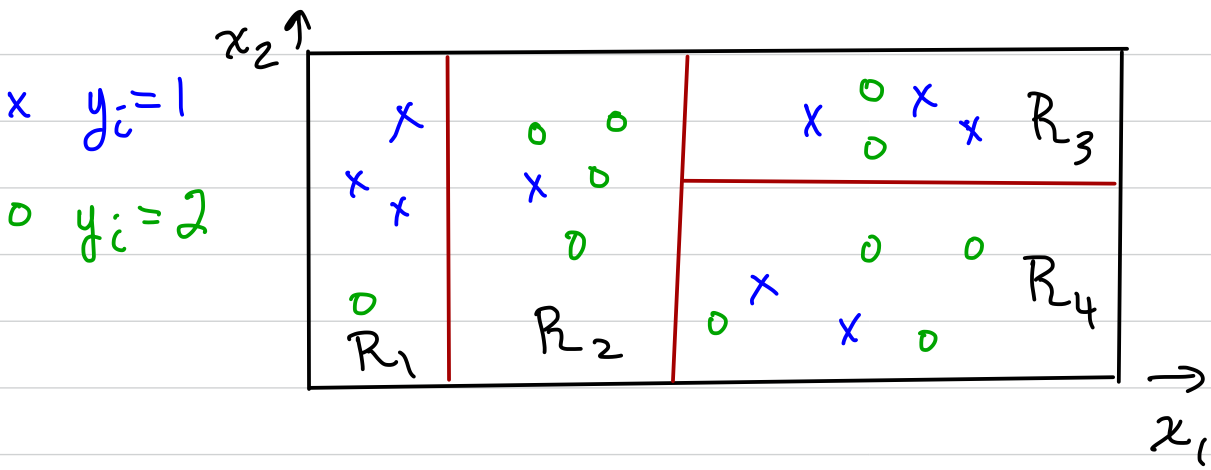
Murphy Fig. 16.1(b)

For classification, a variety of cost functions can be used, e.g.:

$$\text{cost}\{(\underline{x}_i, y_i) \in R_{m'}\} = \frac{1}{N_{R_{m'}}} \sum_{\underline{x}_i \in R_{m'}} \mathbb{I}[y_i \neq \underbrace{\hat{y}(R_{m'})}_{\text{class assignment in } R_{m'}}]$$

$$= \text{classification error rate in } R_{m'}.$$

What class assignment minimizes this cost for each region?



Region	$\hat{y}(R_m) = w_m^*$
R_1	1
R_2	2
R_3	1
R_4	2

→ Class with most data pts. in R_m.

Examples of other cost fcns. are given in Discussion 9 and text.

Multiclass classification

CART with these cost functions works for $C > 2$ classes also.
Each iteration still divides one region $R_{m'}$ into 2 regions.

Tip – to save on computation (for regression and classification)

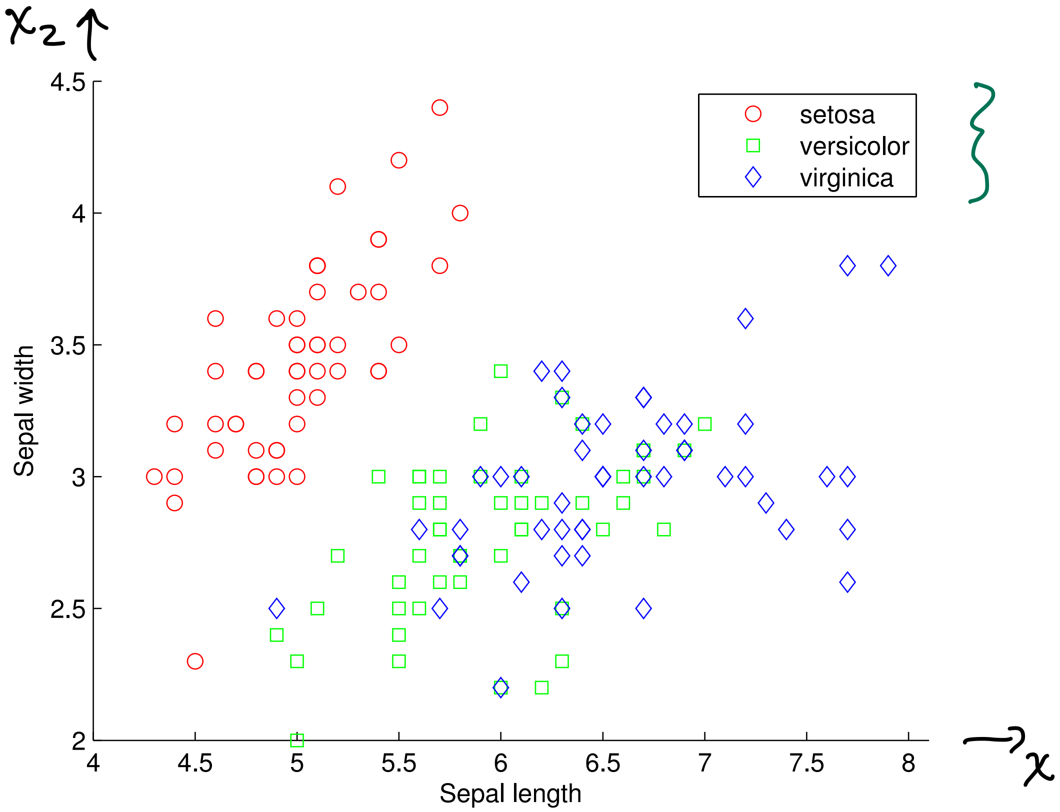
CART typically cycles through all regions R_m, $m = 1, 2, \cdots$, splitting each region into 2 if the halting condition isn't met, instead of finding best region to split at each iteration.

Example: Iris variety classification ($C = 3$)

[Murphy Figs. 16.4–16.5a]

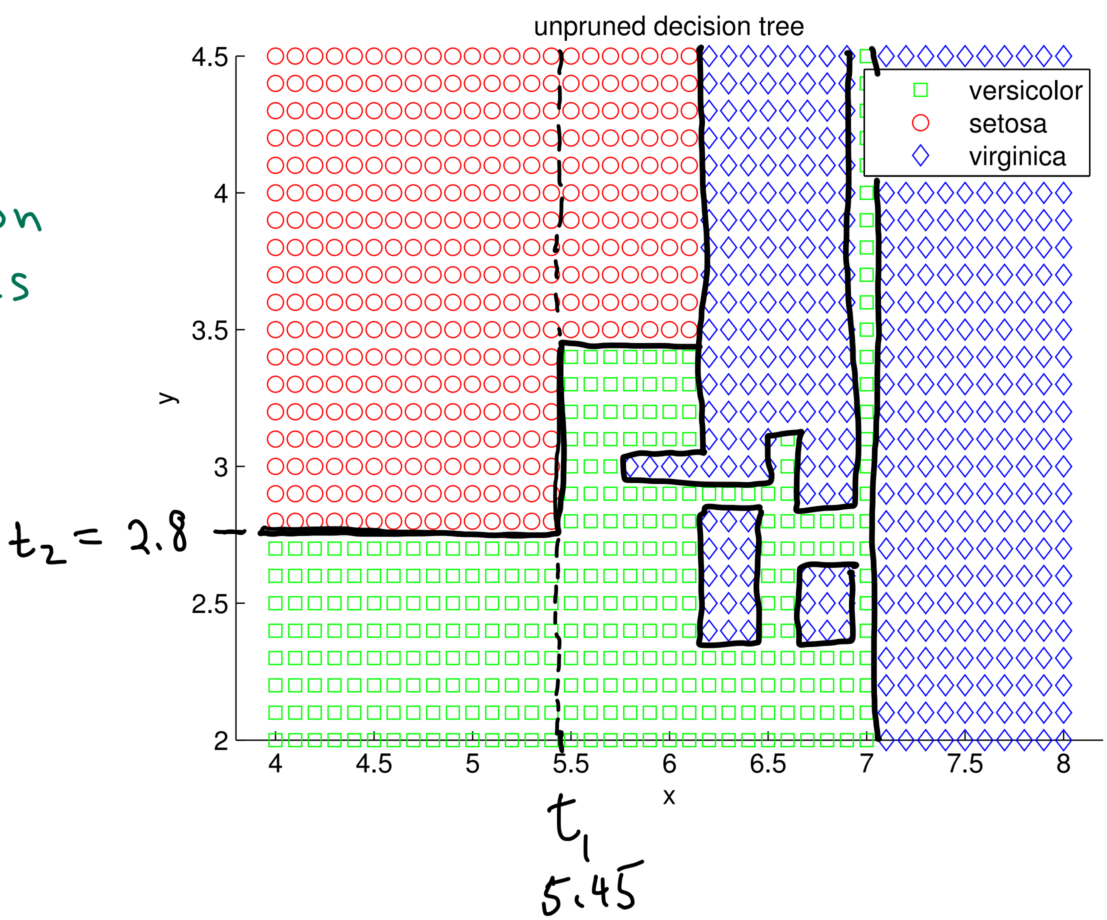
C=3 classes

Iris data

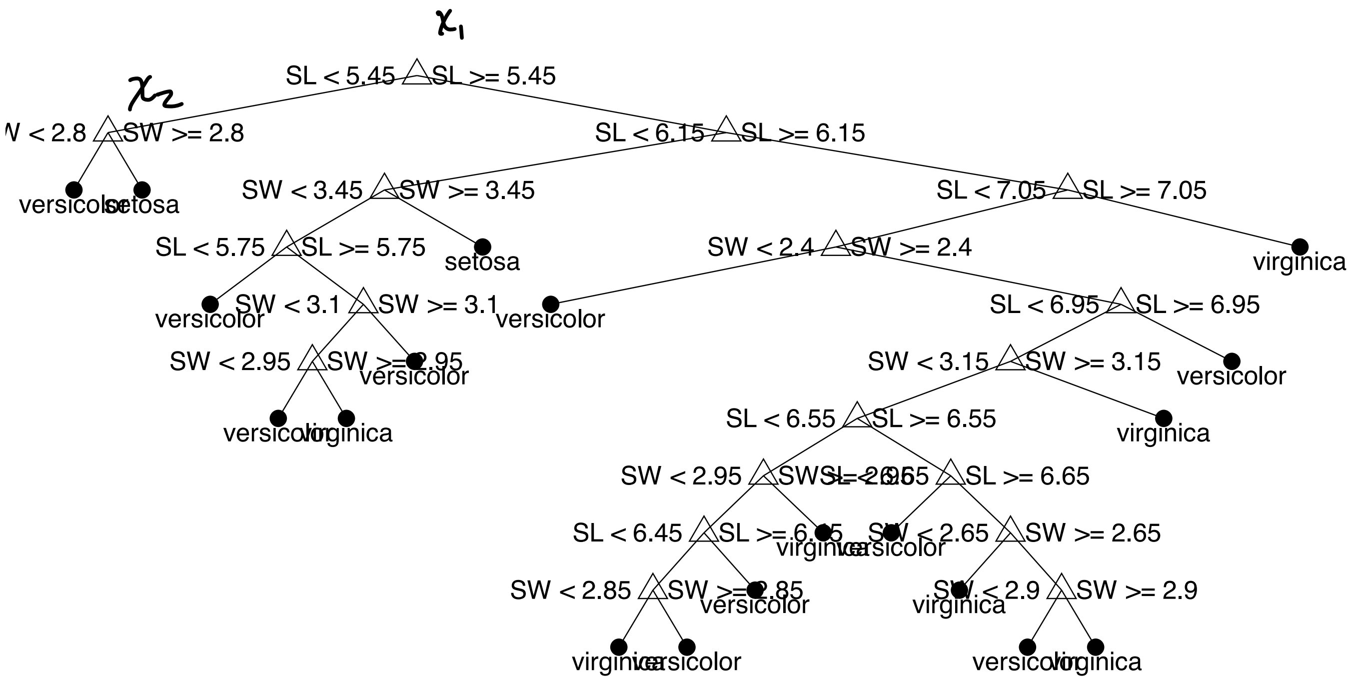


Murphy Fig. 16.4

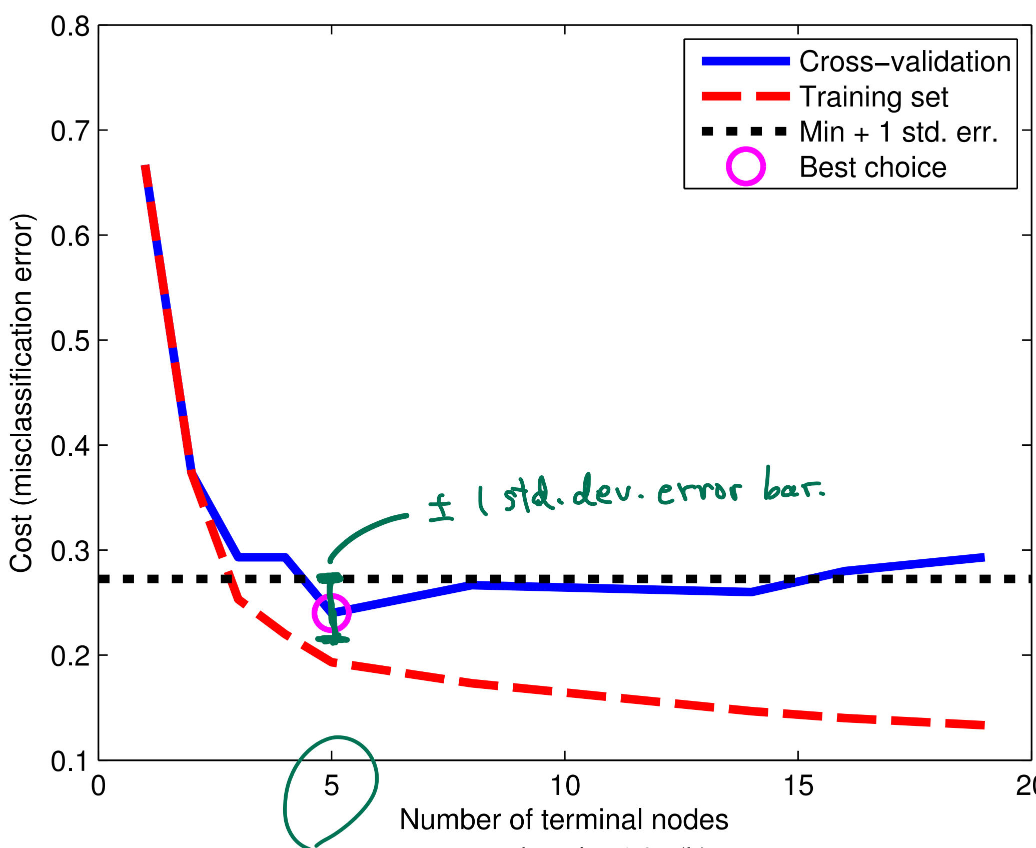
CART Decision regions



CART decision tree



Murphy Fig. 16.5(a)



Murphy Fig. 16.5(b)

- Because CART is a greedy algorithm (does not optimize globally at each iteration), growing the tree until the optimal stopping point typically doesn't yield best results.
- Usually it is run past this point, to yield a tree that overfits.
- Then tree is pruned.

"Weakest link pruning":

1. Collapse the internal node that gives the smallest increase in cost fcn. Iterate.
2. Use cross validation to halt when minimum validation error is reached (within 1σ)*.

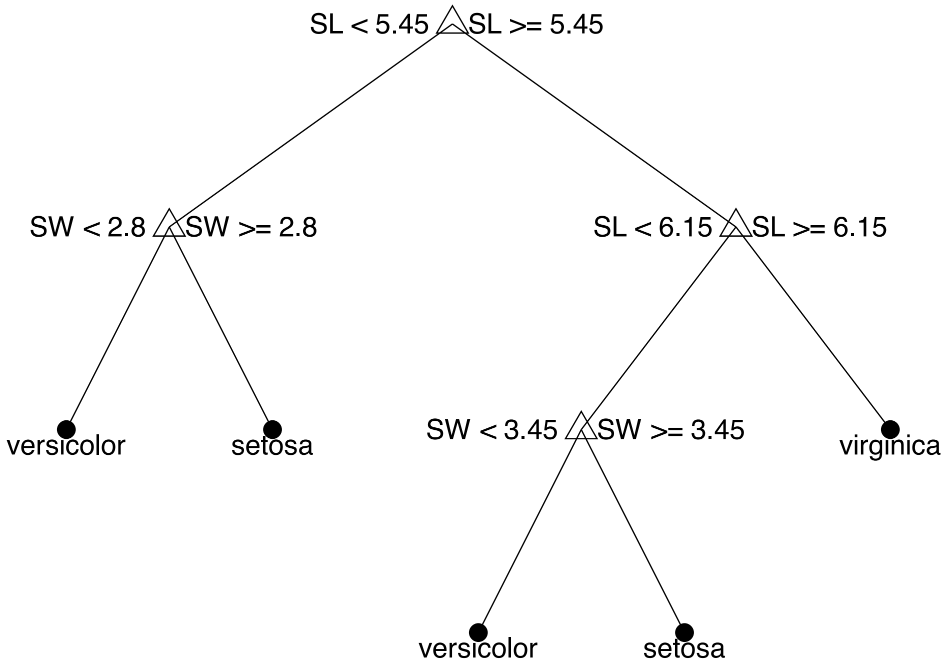
* Picking best tree size from cross validation: A general rule is, among the models that have error $\leq E_{min} + (1$ std. deviation$)$, pick the simplest model.

In the case above $\Rightarrow$ 5 terminal nodes.

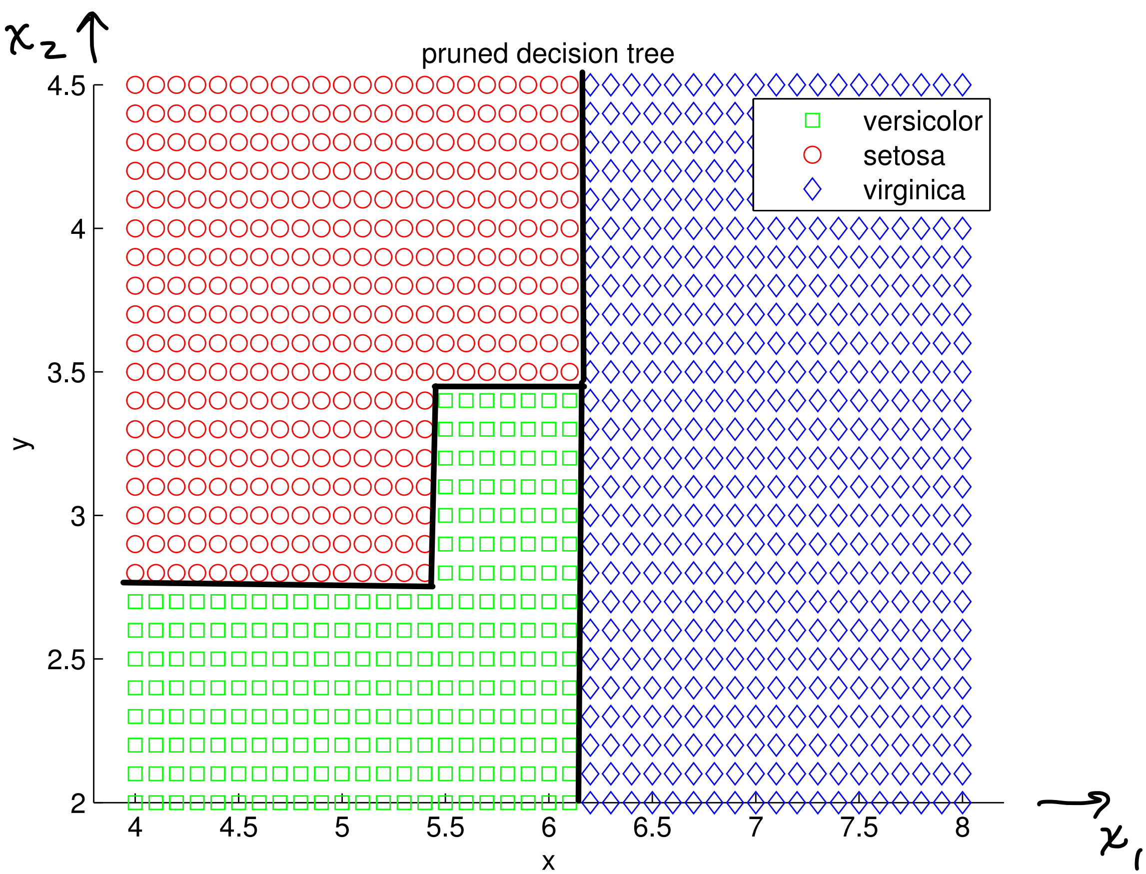
Resulting tree and decision regions in Iris example:

[Murphy Fig. 16.6]

Final decision tree (after pruning) to optimal model:



Decision regions & boundaries



Murphy Fig. 16.6

CART Summary — pro's and cons

Pro's

1. Easy to understand
2. Explainable
3. Easy to adjust complexity (halting condition or size of tree)
4. Can be fast (e.g., small trees)
5. Typically robust to outliers
6. Can give ranking of feature importance, or be used for feature selection.

Cons

1. Easy to overfit
2. May not find a global min. of f_{obj}
3. Decision (or region) boundaries constrained to be parallel to feature axes (each segment)
4. Predictive accuracy often isn't as good as with some other methods.
5. Can be unstable to slight changes in data.
   $\Rightarrow$ high variance.