KD trees and decision trees

James McInerney

Adapted from slides by Nakul Verma

Scaling *k*-NN Classification

- Finding the *k* closest neighbor takes time!
- Need to keep all the training data around during test time!

Speed Issues with *k*-NN

Given a test example $\vec{x}_t$

What is computational cost of finding the closest neighbor?

$O(nd)$

n = # of training data

d = representation dimension

Modern applications of machine learning

n = millions

d = thousands

How can we find the neighbor faster?

Finding the Neighbor Quickly

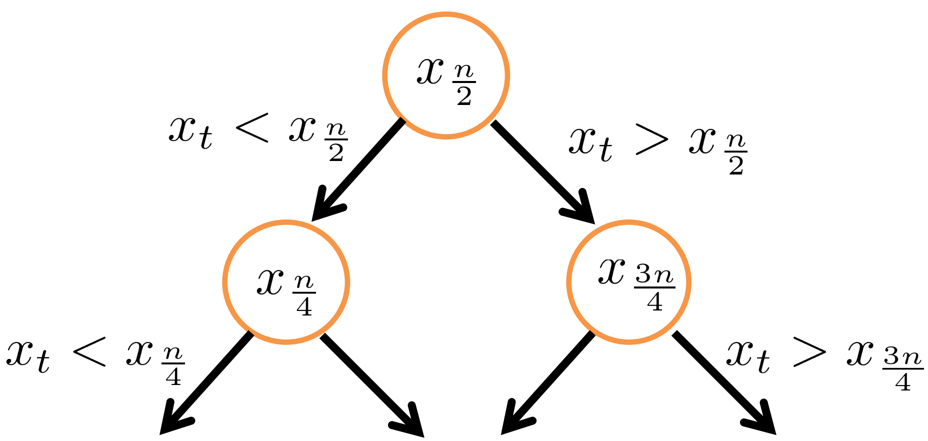
Let's simplify to **R**

How do you find an element x_t from a pool $x_1, x_2, \ldots, x_n$ of examples?

Naïve approach $O(n)$

How can we do the search more quickly?

Say the pool of examples is **sorted**:



Can significantly **improve** the search time

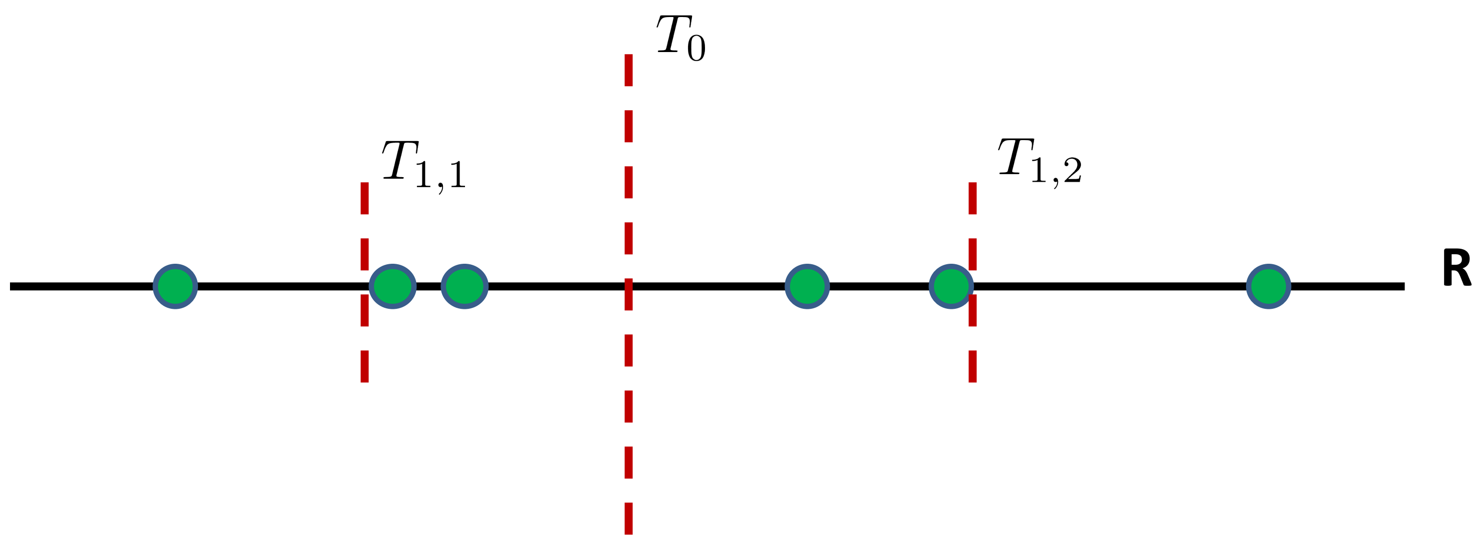
$O(\log n)$

Preprocessing overhead (**sorting**)

$O(n \log n)$

Finding the Neighbor Quickly (contd.)

What if x_t is not in the pool?



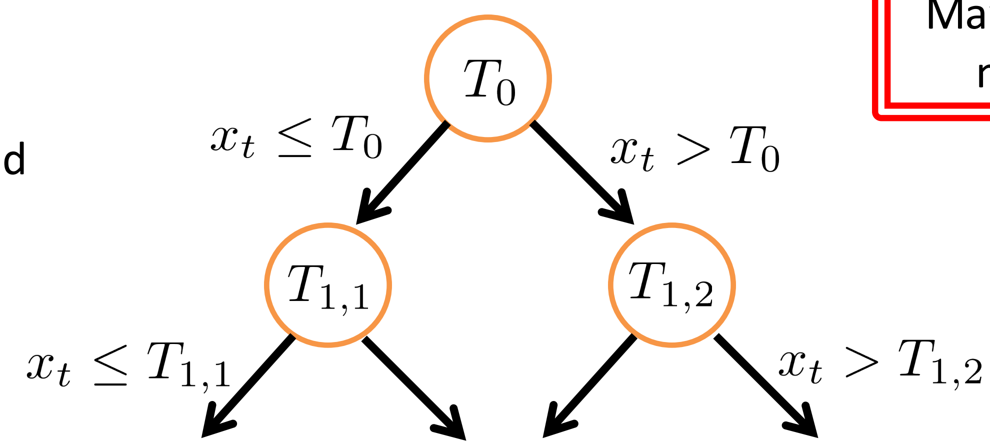
the search time

$O(\log n)$

Preprocessing overhead

(finding **medians**)

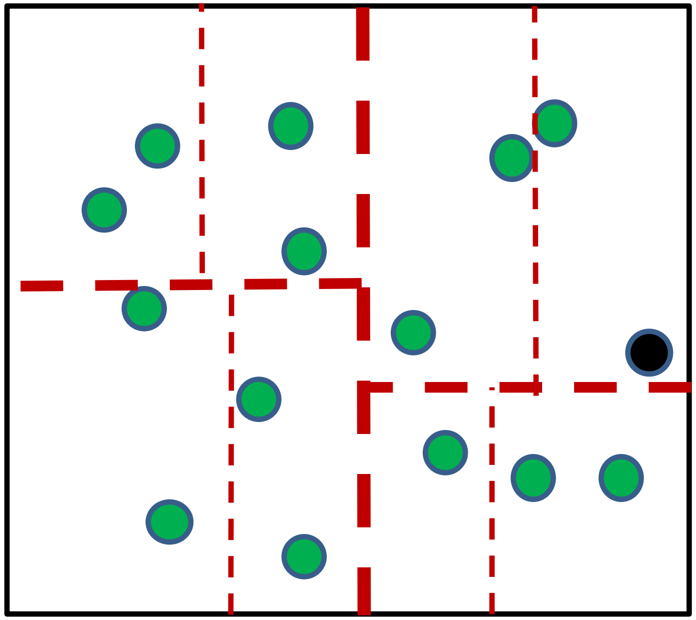
$O(n \log n)$



May not give the exact nearest neighbor!

Finding the Neighbor Quickly (contd. 2)

Generalization to $\mathbf{R}^d$



the search time

$O(\log n)$

Preprocessing overhead
(finding **medians**)

$O(n \log n)$

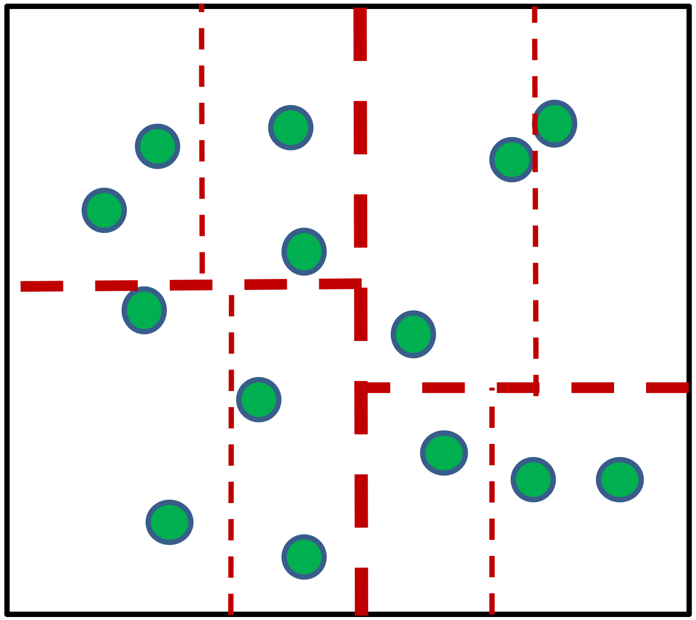
This datastructure is called *k*-d trees

Scaling *k*-NN Classification

- Finding the *k* closest neighbor takes time!
- Need to keep all the training data around during test time!

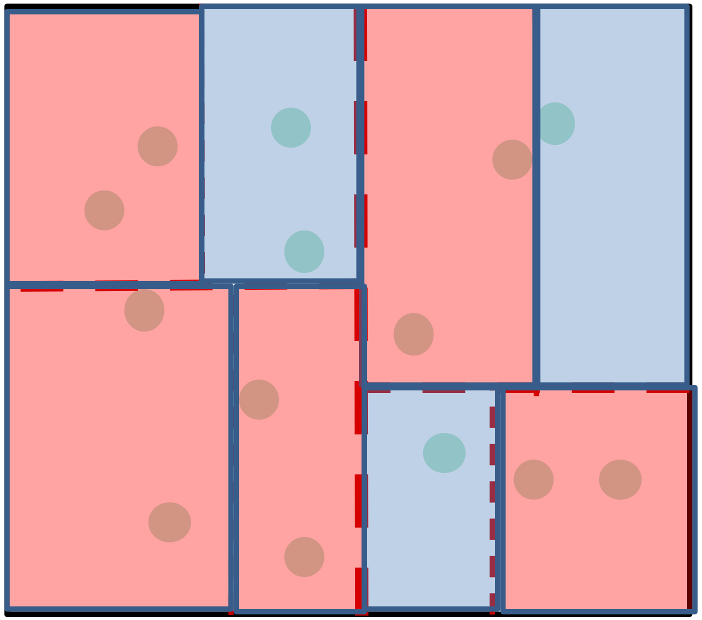
Space issues with *k*-NN

Seems like we need to keep all the training data around during test time



Space issues with *k*-NN

Seems like we need to keep all the training data around during test time

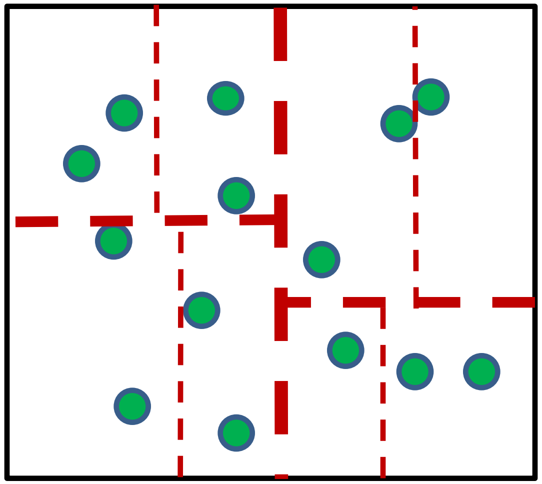


We can **label each cell** instead and discard the training data?

What's the space requirement then? # cells (of width r) = $\min\{n, \approx (1/r)^d\}$

Classification with Trees (Directly)

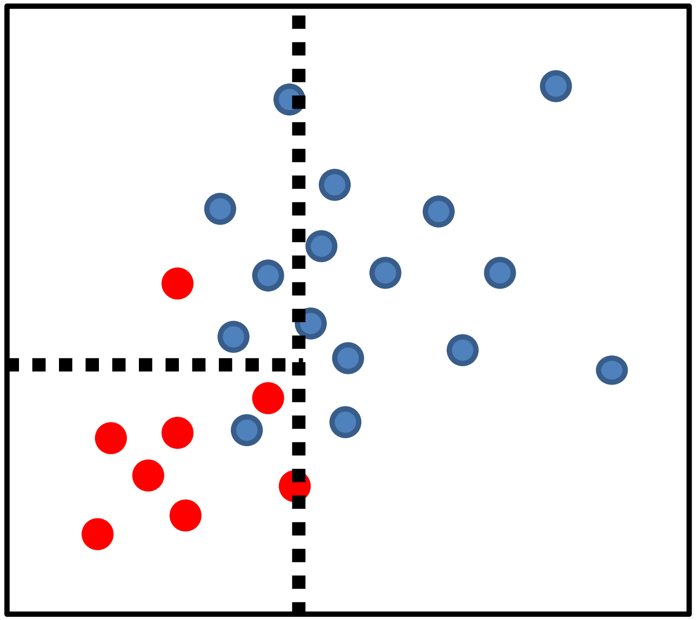
k-d tree construction does not optimize for classification accuracy. Why?



idea: we should choose the features and the thresholds that directly optimize for classification accuracy!

Decision Trees Classifier

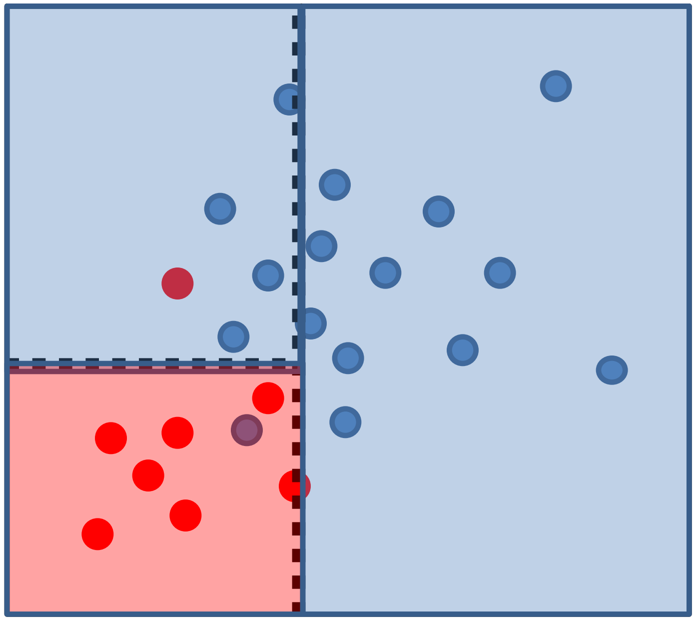
Rather than selecting arbitrary feature and splitting at the median, select the feature and threshold that **maximally reduces label uncertainty**!



done!

Decision Trees Classifier

Rather than selecting arbitrary feature and splitting at the median, select the feature and threshold that **maximally reduces label uncertainty**!



done!

How do we measure label uncertainty?

Measuring Label Uncertainty Cells

Several criteria to measure uncertainty in cell *C*:

classification error: $$u(C) := 1 - \max_y p_y$$

Entropy: $$u(C) := \sum_{y \in \mathcal{Y}} p_y \log \frac{1}{p_y}$$

p_y := fraction of training data labelled *y* in *C*

Gini index: $$u(C) := 1 - \sum_{y \in \mathcal{Y}} p_y^2$$

Thus find the feature *F*, and threshold *T* that **maximally reduces uncertainty**

$$\arg\max_{F,T} \left[u(C) - \big(p_L \cdot u(C_L) + p_R \cdot u(C_R) \big) \right]$$

L = left cell (using *F*, *T*)

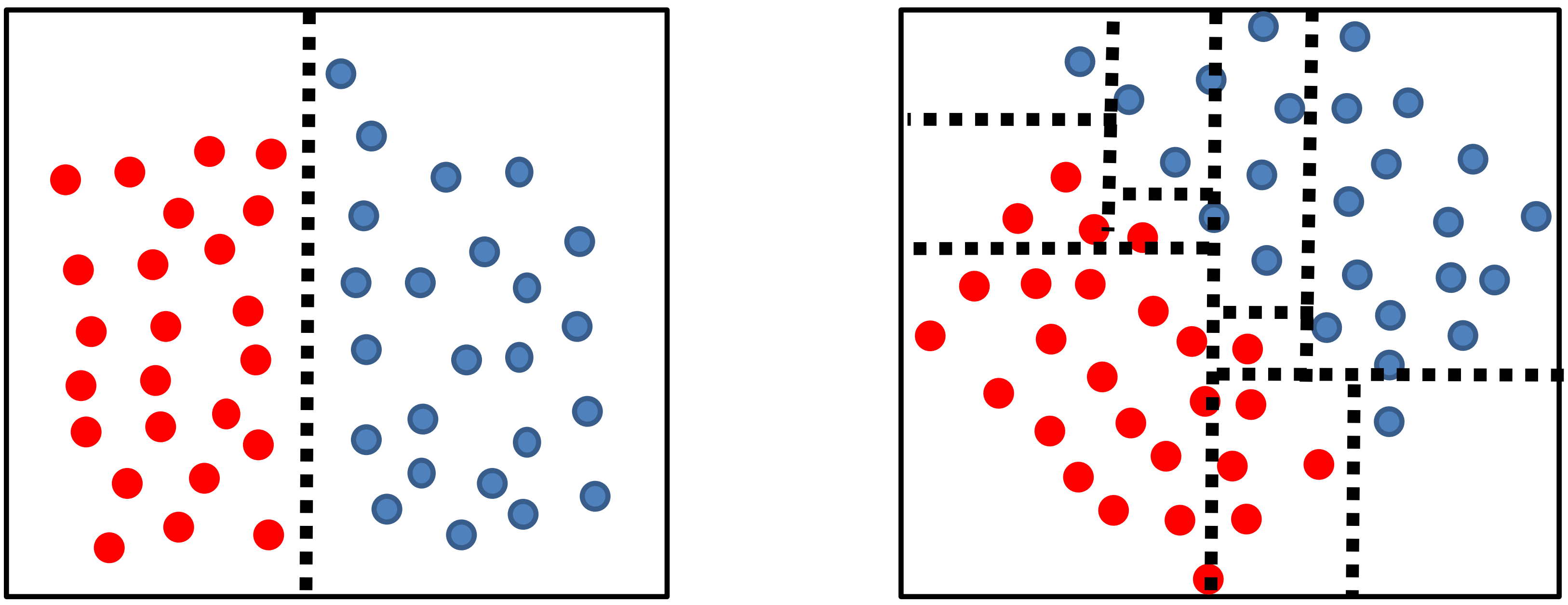
R = right cell (using *F*, *T*)

Decision Tree Observations

- The decision tree construction is via a **greedy approach**
- Finding the optimal decision tree is NP-hard!
- You quickly run out of training data as you go down the tree, so uncertainty estimates become very unstable
- Tree complexity is highly dependent on data geometry in the feature space
- **Popular instantiations** that are used in real-world: ID3, C4.5, CART

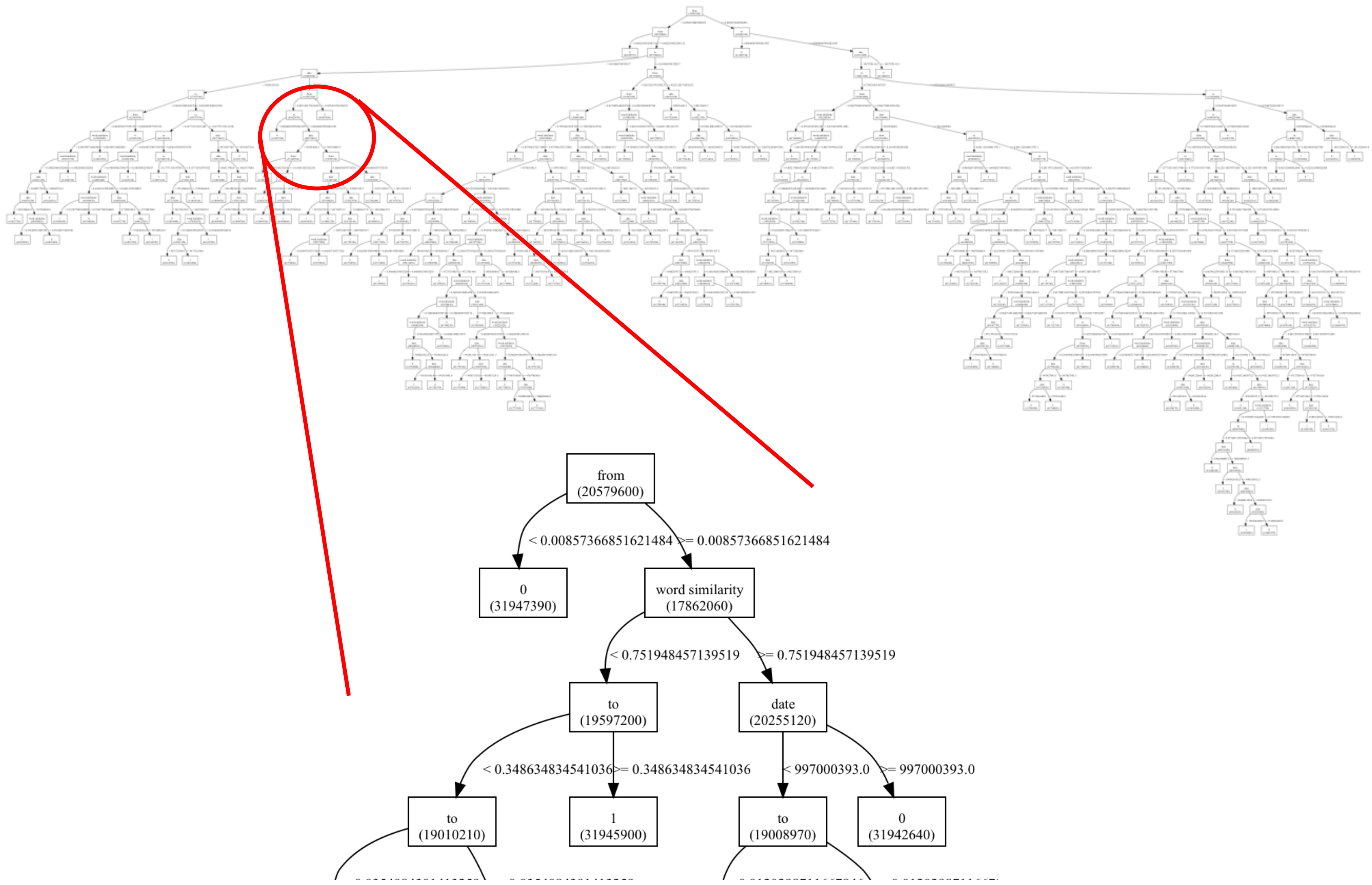
Decision Trees

Tree complexity is highly dependent on data geometry in the feature space

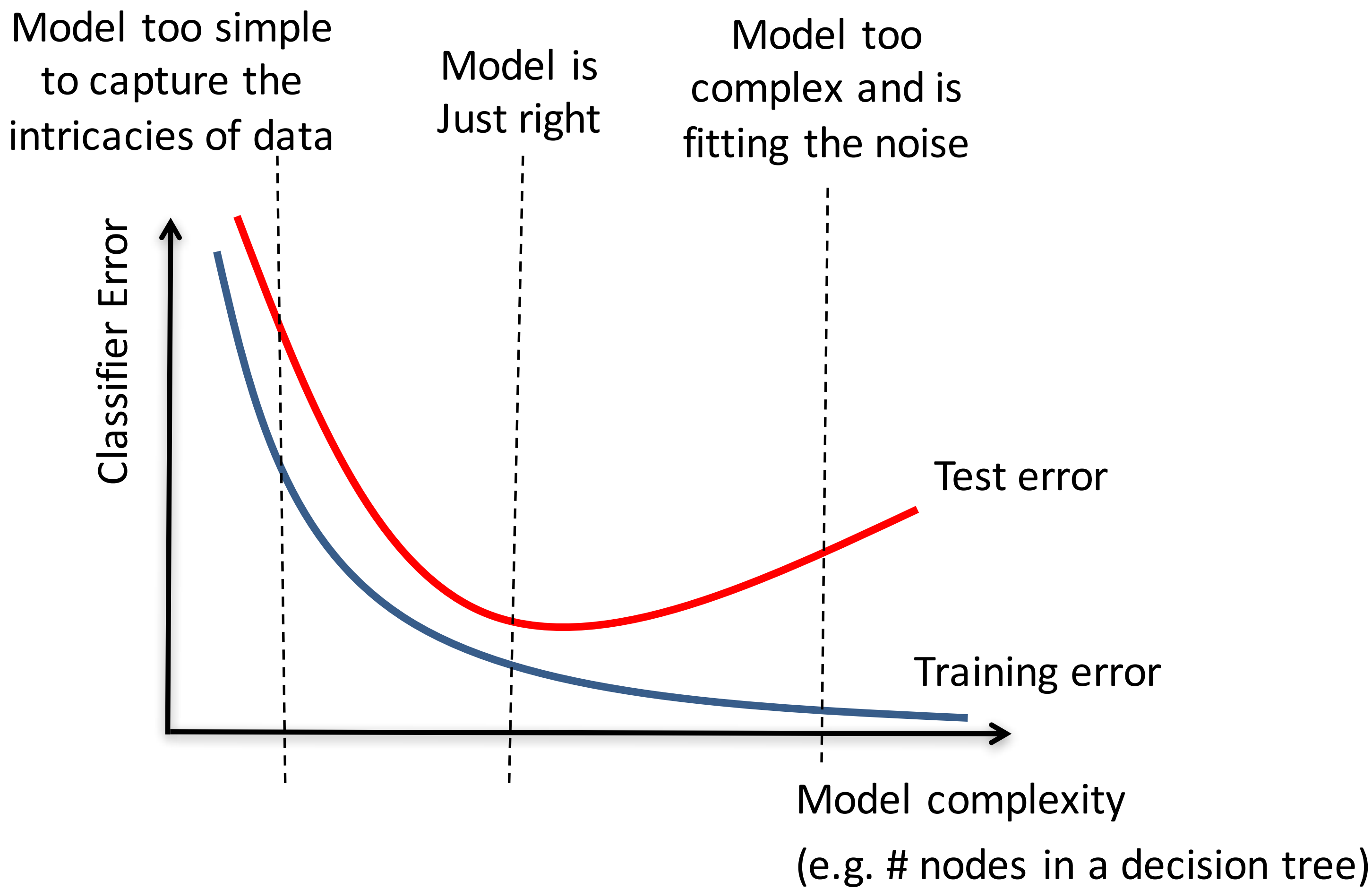


Classifier complexity should not depend on simple transformations of data!

Decision Tree Example (Spam Classification)



Overfitting the training data



How to select a model of the right complexity?

What we learned…

- Coping with drawbacks of *k*-NN
- Decision Trees
- The notion of overfitting in machine learning

Questions?