k-nn and decision trees

k-nn and decision trees don't use GD.

1-nn:

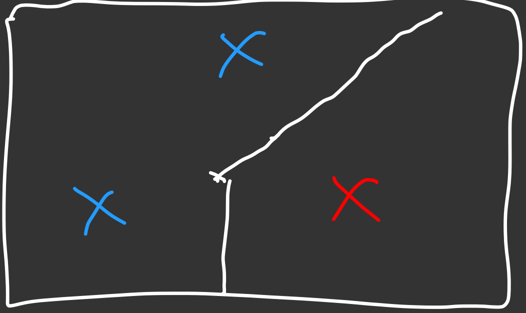
1. pick a distance function
2. memorize training set
3. output closet point

$$\rho(x, x_i) = \min_j \rho(x, x_j)$$

k-nn

1. pick a distance function and integer $k \geq 1$
2. memorize training set
3. classification : output plurety labels among k neis.

   regression : averge .....



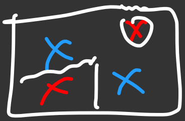
1-nn

3nn: all red
↳ majority

Remarks:

① If $(x_i)_{i=1}^{n}$ $\underbrace{\text{distinct}}_{}$, 1-nn gets

False



0 training error.

② k-nn may fail to get 0 training error.

③ why k-nn.

↳

① higher k $\boxed{\text{smooth}}$ predictor. with less complex model.
(Those random blue labels won't bother)

② Carefully choose k O(nn)

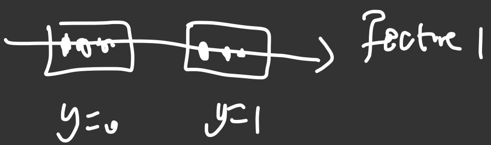
Test error of knn with L_2 distance.

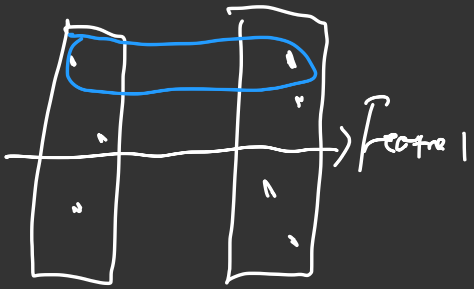
K	1	3	5	7	9
err	0.0309	0.0295	0.0312	0.0306	0.0341

Test error of knn with diff distances

Distance	L_2	L_3	Tangent	shape
err	3.09	2.83	1.10%	0.63%

k-nn can be broken by bad features.





Curse of dimension: Given poly(d) random unit norm points in R^d, with prob > 99%, each is $2 \pm O(1/\sqrt{d})$ from all others.

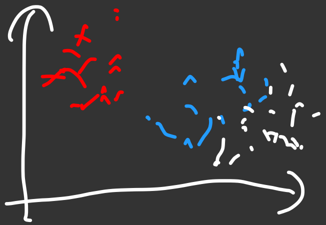
Decision trees

① binary tree which partitions input space.

② each tree node is associated
a splitting rule.

③ leaf node associated with label y.

Traing decision trees



① pick uncertainty measure;

$$u(T) = \frac{1}{n} \sum_{\text{leaf } S \in T} \frac{1}{|S|} u(S)$$

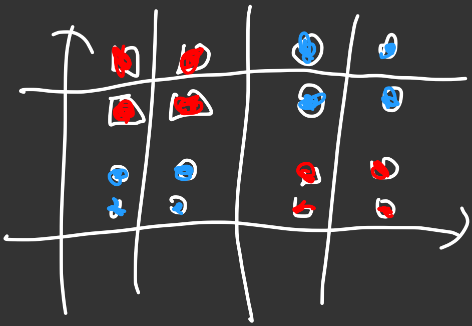
② put all pts at root

③ Loop (thrhold)

  ① pick leaf l and splitting rule h
that maximally reduces uncertainty

  ② split data in l using h and
grow trees accordingly.

The greedy algorithm may fail

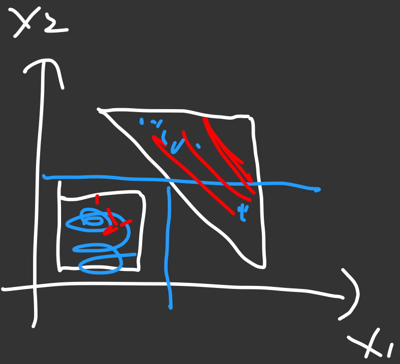


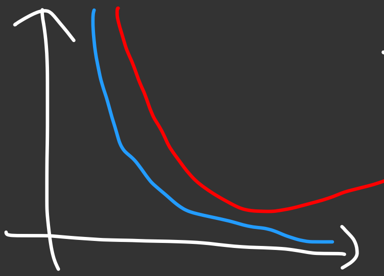
Those ver/hor lines fail half

When to stop

① tree reaches size

② when every leaf is pure (overfitting)





① train error → 0 when tree gets bigger

② test error decreases at first but increases overfitting

Summary

	k-nn	Decision trees.
Train	memorize data	greedy split
Test	find k closest memorized pts	traverse tree output leaf label
overfit	vary k	limit tree size.