AI Theory

Nearest Neighbours

Train : just store D_t

Predictor : $f'(x)$

where distance = min

- 1D - Manhattan
- 2D - Euclidian

Key idea:

Similar examples have similar outputs.

If both distances the same:

$k = 5$

+
−
+
−
+

more + mean

+ is the class.

Training Error:

$$\text{TrainLoss}(w) = \frac{1}{|D_{train}|} \sum_{(x,y) \in D_{train}} \text{Loss}(x,y,w)$$

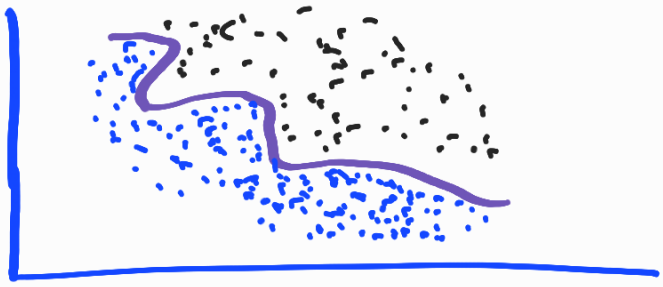
How do we minimize the loss for test data?

Strawman Algo:

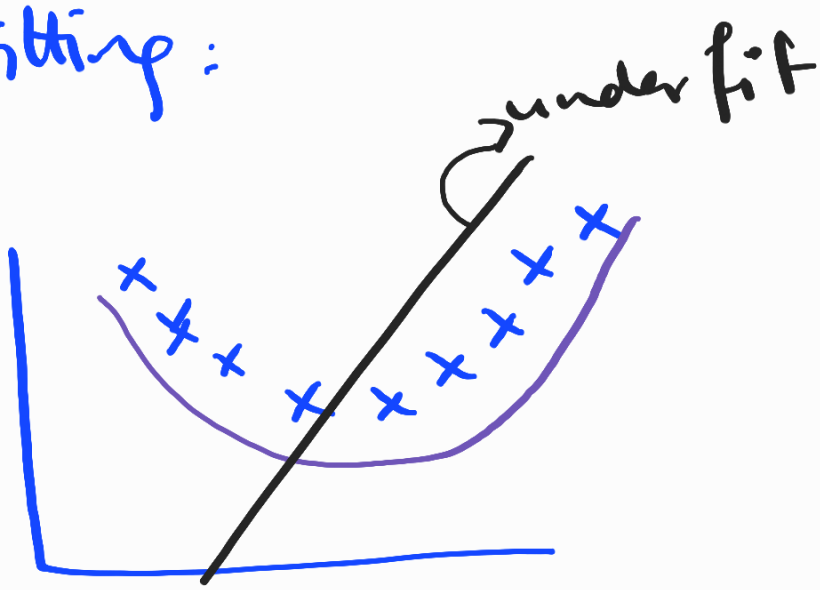
```
Predict(x):
    if x ∈ D_train:
        return y[x]
    return ERROR
```

This is too damn specific.
Also called Overfitting.

Overfitting:



Underfitting:



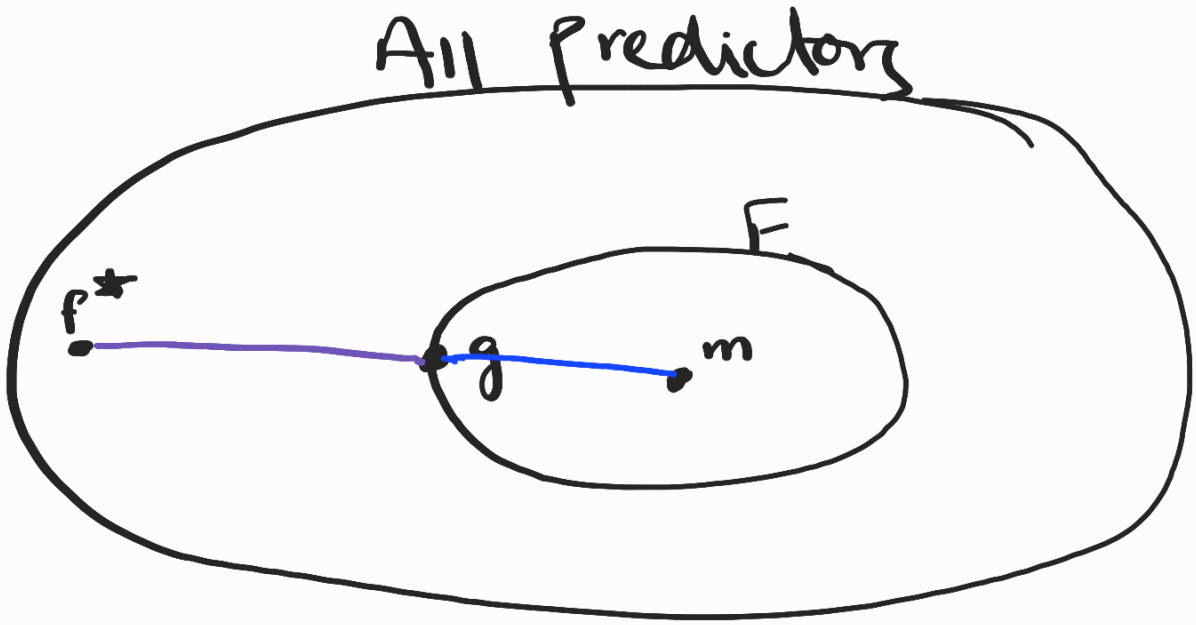
How do we solve this?

Before we solve:

we need to define something.

Approximation & Estimation

Error :



Total Error = Approx + Estimation Error

should be minimum

Dimensionality Reduction:

- Add feature
- Remove feature

- L_2 Regularization
- Forward selection
- PCA

Norm / Weight Decay:

Reduce the length $\|w\|$

(size of the weight)

$0 \longrightarrow 0$

Regularization = $\min\left(\text{Train}^{\text{Loss}}_{(w)}\right)$

$$+ \frac{\lambda}{2} \|w\|^2$$

where λ is the penalty for training loss.

in gradient Descent, update weights as:

$$w \leftarrow w - \alpha\left(\frac{\partial \text{Loss}}{\partial w} + \lambda w\right)$$

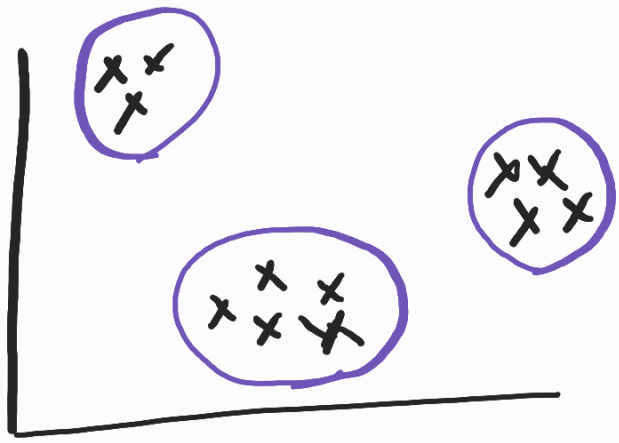
Hyperparameters (learn yourself)

Unsupervised Learning

Clustering

what the name shows:

GROUP BY data.



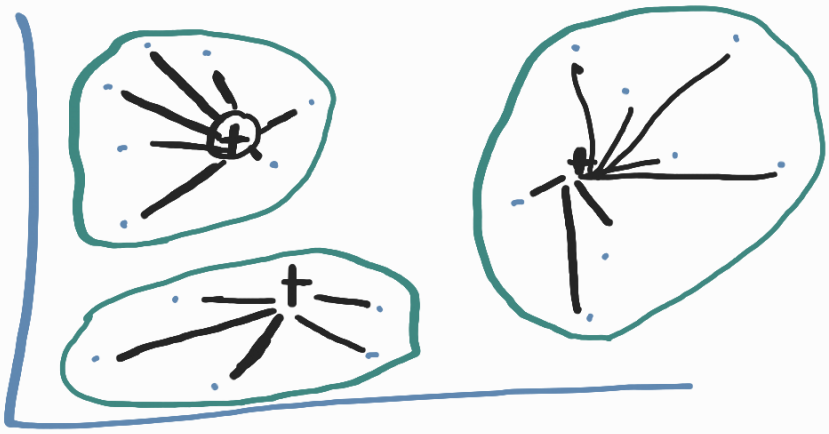
KMeans Clustering:

→ How many clusters to make?

Setup:

↳ Each cluster is a centroid.

↳ want each point to be close to its assigned centroid.



Dry run on 1D data.

Algorithm:

1. min distance to centroid results in regrouping

2. Take average of all points and make a new centroid. Repeat until convergence.

Heirarchical clustering.