SL04. Instance Based Learning

Introduction:

- Normal ML algorithms uses input data (x, y) and searches the hypotheses space for the best generalized function $f(x)$ that can be used to predict new values.
- In Instance Based Learning, we create a database of all x/y relationships, and once we receive a new value x we lookup this database to find its corresponding y.
- Advantages:
  - The database is the saves the exact values of y not an approximation of it.
  - This way is fast. No need for learning.
  - It's simple.
- Disadvantages:
  - No generalization.
  - Prone to overfitting:
    1. Models noise.
    2. Lookup can return multiple values for the same input point.

k-Nearest Neighbors:

- This algorithm simply looks at a selection (k → free parameter) of the nearest neighbors of the query point and selects the best majority label.
- This is how it works:
  - Given:
    1. Training data (D).
    2. Distance (similarity) metric $d(q, x)$ → domain knowledge.
    3. Number of neighbors (k) → domain knowledge.
    4. Query point (q).
  - Find:
    1. A set of nearest neighbors (NN) such that $d(q, x)$ is smallest.
  - Return:
    1. Classification: We perform a vote to select y_i of NN, where y_i is the most frequent (Highest plurality - Mode).
    2. Regression: Compute the mean of all y_i.
    3. We can also use a weighted vote of weighted average, which means that the closer the i point is to the query point q, the more influence its y_i has on the vote/mean.

- Comparison:

Algorithm		Running Time	Space
$1-\text{NN}$	Learning	1	n
	Query	$\log n$	1
$k-NN$	Learning	1	n
	Query	$\log n + k$	1
Linear Regression	Learning	n	1
	Query	1	1

  - Lazy vs Eager learners: The NN learning algorithm is a lazy learner. It procrastinates learning till it starts querying. On the other hand, Linear Regression is an Eager learning algorithm, performing all learning initially to come up the best hypothesis.

$k-NN$ Preference Bias:

- Locality:
  - Points that are nearby are similar.
  - This concept is embedded in the distance function.
  - There will be distance functions that can model the given problem and some that can't. Choosing a suitable distance function requires domain knowledge.
- Smoothness: Average produces smoothness as opposed to discontinuity.
- All features matter **equally**.

Curse of Dimensionality:

- As the number of features or dimensions grows, the amount of data that we need to generalize accurately grows exponentially.
- For any Machine Learning algorithm, it might be logical to add more features (dimensions) to determine which feature is actually important. But as you add more features, you need to add more data.
- One way to deal with the dimensionality issue is to give a different weight to each dimension.

Conclusion:

- For a distance metric, we can use any function to calculate distance (e.g. Euclidean, Manhattan, weighted-components, mismatches, etc.).
- Implications of the value of k:
  - If $k = n$ we'll end up with a constant average function. But if we used weighted average, the points closer to the query point will have greater influence.
  - We can also use other methods instead of weighted-average. We can use "local" linear regression on the nearest k points, this is known as Locally Weighted Regression"
  - We can even use Decision Trees, Neural Networks, etc. to come up with more complicated hypotheses (More representative curves). This, though, might cause overfitting.