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
$1-NN$	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

Eager vs Lazy Learners:

- Eager learners:
  - We infer a function that best fits our training data.
  - When we have new inputs, the input's features are fed into the function, which produces an output.
  - Once a function has been computed, the data could be lost without affecting the model's performance (until we obtain more data and want to re-compute our function).
  - We take the time to learn from the data first and sacrifice local sensitivity to obtain quick, global scale estimates on new data points.
- Lazy learners:
  - We don't compute a function to fir the training data before new data is received.
  - New instances are compared to the training data itself to make a classification or regression decision.
  - Memory requirements are much larger than eager learners (storing all training data versus just a function).
  - Taking decisions on new examples takes longer to compute than eager learners.
  - Lazy learners have advantages in local scale estimation and easy integration of additional training data.
- 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:

- Implications of the choice of d:
  - We can use any function to calculate distance (e.g. Euclidean, Manhattan, weighted-components, mismatches, etc.).
  - Weights may be placed on certain factors if there is reason to believe that these factors are more or less important to classification or regression (The distance function should remain a valid distance metric).
  - Reducing the weight on factors can also help reduce the effects of dimensionality on the algorithm's performance.
  - The measure of closeness does not need to be a simple mode or mean. Averages may utilize weights on training instances as a function of their distance or similarity to the test instance.
- Implications of the value of k:
  - When k is small, models have high variance, fitting on a strongly local level.
  - Larger k creates models with lower variance but higher bias, smoothing out the influence of individual data points on output decisions. If $k = n$ we'll end up with a constant average function.
  - 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.