# Ethan Hadley

# 1:

Algorithms generally work by having a set of clusters defined by their centers, or centroids. The iteratively get shifted around to lie within morem tightly assocaited datapoints. Naive methods often proceed as:

- Randomly select k datapoints, where k is the number of desired clusters, and selecting them as the centroid starting points.
- For each datapoint, assign that datapoint to the cluster it is currently nearest to.
- The cluster's position then moves, becoming the mean position of all the datapoints attributed to it.
- Repeat until all datapoints are assigned.

# 2:

An SVM's optimization goal is to describe a plane in the possibly high dimensional space of the dataset, such that the plane divides dataset elements most cleanly based on their class. By 'cleanly' what is meant is having the largest minumum gap between any two elements of different class. The plane which has the largest distances between different classed datapoints is the Maximum Marginal Hyperplane.

# 3:

A k-d tree is a datastructure which splits the space into smaller regions, partitioning the dataset based on its position in the space. Selecting good partitions, this allows us to search through the tree much faster than searching through the full dataset. This makes operations like 'find the nearest neighbor to this new datapoint' much faster on a k-d tree.

# 4:

Eager learning systems attempt to compress or algorithmically describe the patterns (which are the same thing) inherent in the data they are trained on. Lazy learning involves simply using the entire dataset as is, using similarity measures to evaluate the properties of new datapoints which require classification, prediction etc.

The advantages of eager learning systems is that they typically (for reasonable dataset/model sizes) require less space to store than the entire dataset they were trained on. The heavier computation required for an eager system is only required once, at training time. The disadvantages is that they are typically lossy approximators (information present in the dataset can be lost, not transferred to the model).

The advantages of lazy systems is that they require relatively less computation during training, and that they for the most part contain all of the data present in the original dataset, without loss of information (but finding methods which *apply* all of the information present for new datapoints is difficult). The disadvantages is that they require storing and loading the whole dataset at inference time. As a result, the computational cost at inference time of lazy learning systems is usually larger than that of eager systems.