The k-NN Algorithm

Assumption: Similar inputs have similar outputs

Classification Rule: For a test input $\underline{x}$, assign the most common label amongst its k most similar training inputs

Formal (and borderline incomprehensible) **Definition:** Test point $\underline{x}$

Denote set of k-nearest neighbors of $\underline{x}$ as $S_{\underline{x}}$.

Formally,

$$S_{\underline{x}} \subseteq D \text{ s.t. } |S_{\underline{x}}| = k \text{ and } \forall (\underline{x}', y') \in D \setminus S_{\underline{x}},$$
$$\text{dist}(\underline{x}, \underline{x}') \geq \max_{(\underline{x}'', y'') \in S_{\underline{x}}} \text{dist}(\underline{x}, \underline{x}'')$$

We can then define a classifier $h()$ as a function returning the most common label in $S_{\underline{x}}$:

$$h(\underline{x}) = \text{mode}(\{y'' : (\underline{x}'', y'') \in S_{\underline{x}}\})$$

where mode(.) is selecting highest occurrence

What Distance Function?

The better that metric reflects label similarity, the better the classifier will be.

Most common choice is the **Minkowski distance**.

$$\text{dist}(\underline{x}, \underline{z}) = \left(\sum_{r=1}^{d} |[x]_r - [z]_r|^p \right)^{1/p}$$

$[x]_r$: r^{th} dimension vector

$p=1$; Manhatten

$p=2$; Euclidean

$p=\infty$; Max Difference of 2 dimensions

Begin Digression

Bayes Optimal Classifier

Example: Assume you knew $P(y \mid \underline{x})$, then you would simply predict the most likely label.

The Bayes optimal classifier predicts: $y^* = h_{opt}(\underline{x}) = \underset{y}{\operatorname{argmax}} \; P(y \mid \underline{x})$

Although Bayes optimal classifier is as good as it gets, it can still make mistakes. It is **ALWAYS** wrong if a sample does not have the most likely label.

We can compute the probability of that happening precisely (which is exactly the error rate):

$$\varepsilon_{BayesOpt} = 1 - P(h_{opt}(\underline{x}) \mid y) = 1 - P(y^* \mid \underline{x})$$

Assume an email $\underline{x}$ can be classified as **spam** $(+1)$ or **ham** (-1). For the same email $\underline{x}$, the conditional class probabilities are:

$$P(+1 \mid \underline{x}) = 0.8$$
$$P(-1 \mid \underline{x}) = 0.2$$

In this case the Bayes optimal classifier would predict the label $y^* = +1$ [as it is most likely], and its error rate would be $\varepsilon_{BayesOpt} = 0.2$

↑ Bayes optimal classifier → highly informative lower bound of error rate

NOTHING DOES BETTER

Best Constant Predictor

- Essentially predicts always the same constant independent of any feature vectors
- The best constant is the most common label in the training set
  - If k=n, the k-NN becomes this

In regression settings, the best constant is the constant that minimizes the loss on the training set

- for squared loss: average label
- for absolute loss: median label

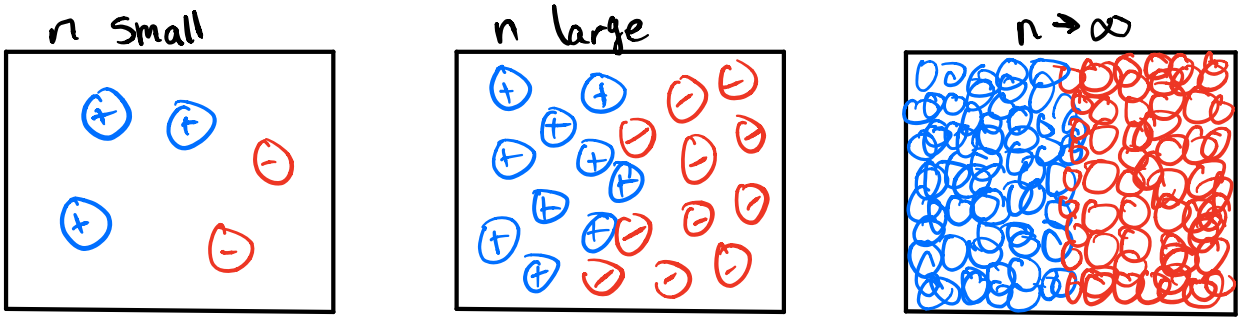
YOU SHOULD ALWAYS BE ABLE TO SHOW THAT YOUR CLASSIFIER PERFORMS SIGNIFICANTLY BETTER ON THE TEST SET THAN THE BEST CONSTANT!

i.e Best Constant Predictor is an upper bound on the error

END DIGRESSION

1-NN Convergance Proof

As $n \to \infty$, the 1-NN error is no more than twice the error of the Bayes Optimal classifier (similar guarantees hold for $k > 1$)



Let $\underline{x}_{NN}$ be the nearest neighbor of test point $\underline{x}_t$.

As $n \to \infty$, $\text{dist}(\underline{x}_{NN}, \underline{x}_t) \to 0$; i.e $\underline{x}_{NN} \to \underline{x}_t$ ← this means nearest neighbor is identical to x_t

You return the label of $\underline{x}_{NN}$.

What is the probability that this is **NOT** the label of $\underline{x}_t$?

(This is the probability of drawing two different labels of $\underline{x}$)

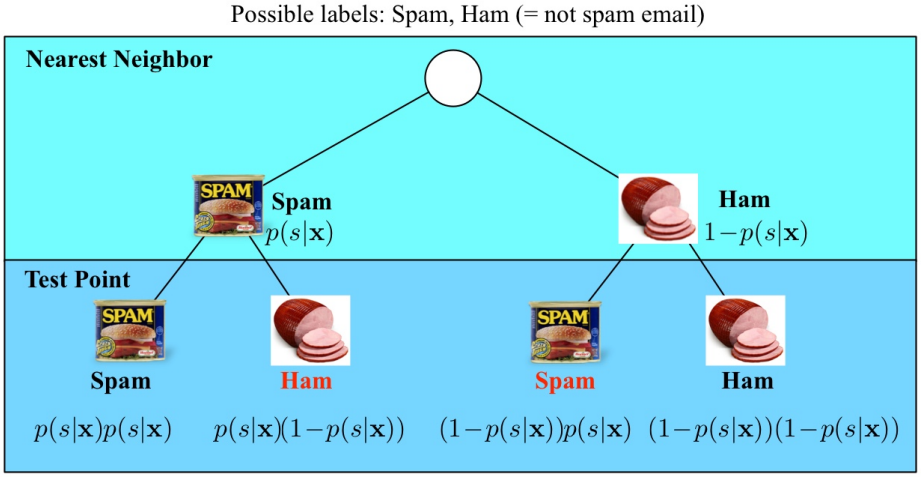
$$\varepsilon_{NN} = \mathbb{P}(y^*|\underline{x}_t)(1-\mathbb{P}(y^*|\underline{x}_{NN})) + \mathbb{P}(y^*|\underline{x}_{NN})(1-\mathbb{P}(y^*|\underline{x}_t))$$

$$\leq (1-\mathbb{P}(y^*|\underline{x}_{NN})) + (1-\mathbb{P}(y^*|\underline{x}_t)) \qquad \mathbb{P}(y^*|\underline{x}_t) \leq 1 \text{ ; } \mathbb{P}(y^*|\underline{x}_{NN}) \leq 1$$

$$= 2(1-\mathbb{P}(y^*|\underline{x}_t)) \qquad \mathbb{P}(y^*|\underline{x}_{NN}) = \mathbb{P}(y^*|\underline{x}_t)$$

$$= 2\varepsilon_{BayesOpt}$$

Example:



In the limit case, the test point and its nearest neighbor are identical.

There are exactly two cases when a misclassification can occur: when the test point and its nearest neighbor have different labels.

The probability of this happening is the probability of the two red events!

$$(1-\mathbb{P}(s|\underline{x}))\,\mathbb{P}(s|\underline{x}) + \mathbb{P}(s|\underline{x})(1-\mathbb{P}(s|\underline{x}))$$

$$= 2\mathbb{P}(s|\underline{x})(1-\mathbb{P}(s|\underline{x}))$$

Good news: As $n \to \infty$, the 1-NN classifier is only a factor of two worse than the best possible classifier.

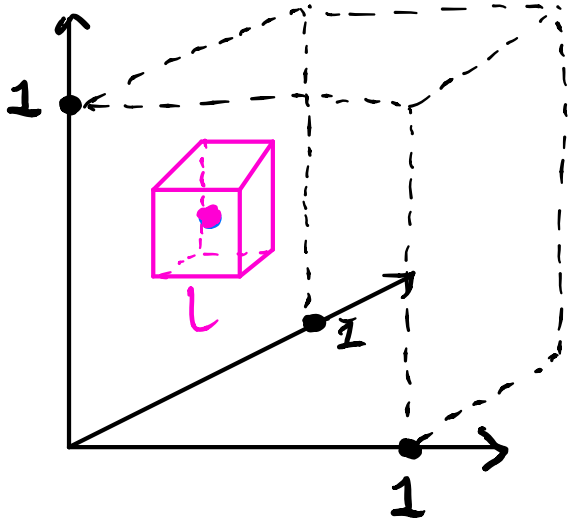
Bad news: We are <u>cursed!</u>

Curse of Dimensionality

k-NN classifier makes the assumption that similar points share similar labels.

In high dimensional spaces, points that are drawn from a probability distribution, tend to never be close together.

Example: We will draw points uniformally at random w/in the unit cube and we will investigate how much space the k-NN of a test point inside this cube will take up.



Formally, imagine the unit cube $[0,1]^d$. All training data is sampled uniformly w/in this cube. i.e $\forall\, x_i \in [0,1]^d$ and we are considering the $k=10$ nearest neighbors of such a test point.

Let l be the edge length of the smallest hypercube that contains all k-nearest neighbors of a test point.

Then

$$l^d \approx \frac{k}{n} \quad \text{and} \quad l \approx \left(\frac{k}{n}\right)^{1/d}$$

If $n = 1000$, how big is l?

d	l
2	0.1
10	0.63
100	0.955
1000	0.9954

So as $d \gg 0$ almost the entire space is needed to find the 10-NN.

This breaks down the k-NN assumptions!

- b/c k-NN are not particularly closer than any other data points in the training set

Why not just increase amount of training samples n until nearest neighbors are close to ~~test~~ point?

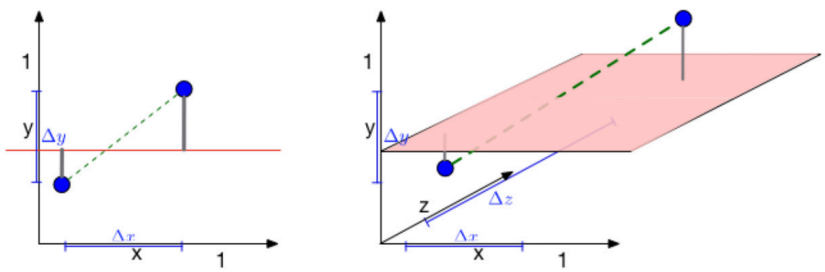
Fix $l = \frac{1}{10} \Rightarrow n = \frac{k}{l^d} = k \cdot 10^d \Rightarrow$ for $d > 100$ need more points then electrons in universe

Distances to Hyperplanes

So the distance between two randomly drawn datapoints increases drastically w/ dimensionality.

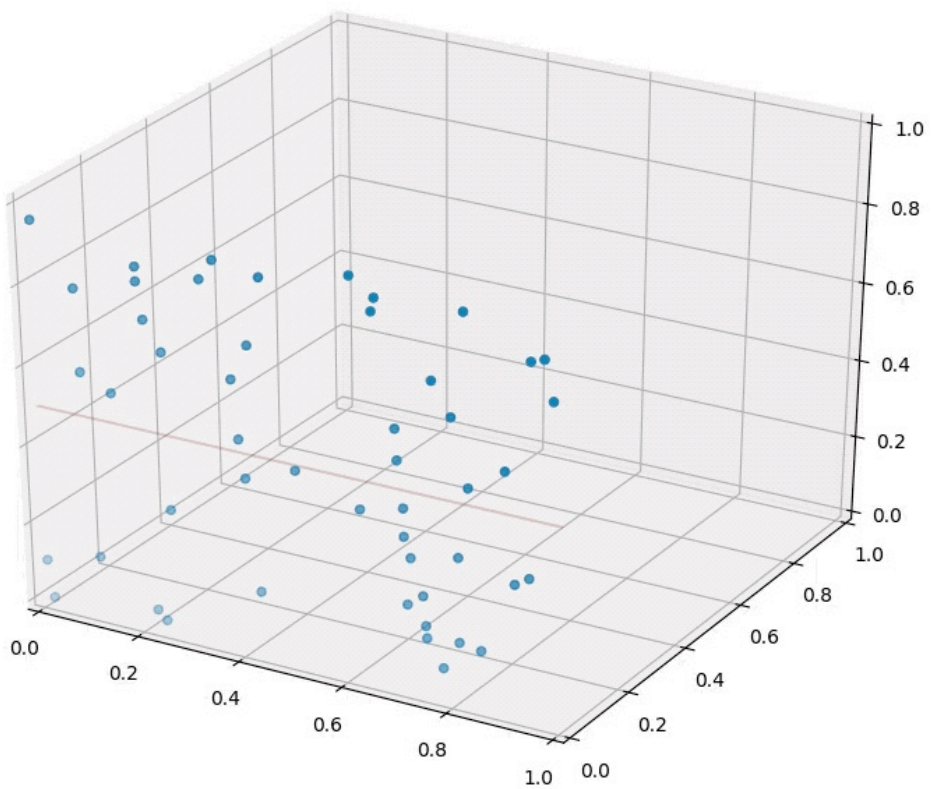
What about distances to hyperplanes?

As long as $d=2$, $d(p_1, p_2) = \sqrt{\Delta x^2 + \Delta y^2}$. If we add an dimension, this extends to $\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$ which is at **least** as large. But the distance to the hyperplane remains unchanged! Why? The normal of the hyperplane is orthogonal to the new dimension!



The curse of dimensionality has different effects on distances between two points and distances between points and hyperplanes.

So in d dimensions, d-1 of them are orthogonal to the normal of any given hyperplane. Movement in those dimensions cannot increase or decrease the distance to the hyperplane – the points just shift and remain at the same distance.

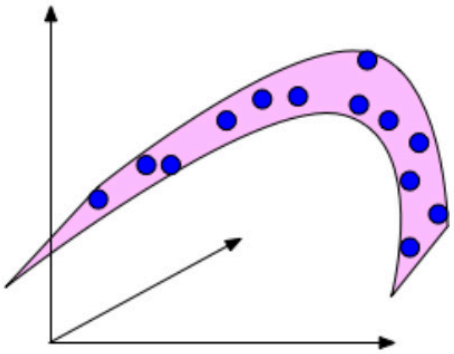


An animation illustrating the effect on randomly sampled data points in 2D, as a 3rd dimension is added (with random coordinates). As the points expand along the 3rd dimension they spread out and their pairwise distances increase. However, their distance to the hyper-plane (z=0.5) remains unchanged --- so in relative terms the distance from the data points to the hyper-plane shrinks compared to their respective nearest neighbors.

A consequence w/ curse of dimensionality is most points tend to be very close to these hyperplanes and it is often possible to perturb the input slightly in order to change a classification outcome.

Data with Low Dimensional Structure

However, not all is lost. Data may lie in low dimensional subspace or on sub-manifolds. Example: natural images (digits, faces). Here, the true dimensionality of the data can be much lower than its ambient space. The next figure shows an example of a data set sampled from a 2-dimensional manifold (i.e. a surface in space), that is embedded within 3d. Human faces are a typical example of an intrinsically low dimensional data set. Although an image of a face may require 18M pixels, a person may be able to describe this person with less than 50 attributes (e.g. male/female, blond/dark hair, ...) along which face vary.



An example of a data set in 3d that is drawn from an underlying 2-dimensional manifold. The blue points are confined to the pink surface area, which is embedded in a 3-dimensional ambient space.