Outline

Learning Vector Quantization

- Though K-means clustering helps to significantly reduce the memory load of k-NN, the most obvious shortcoming of this method is that prototypes from different classes have no say about each other's positioning.
- Recalling the VQ learning algorithm from clustering techniques for unsupervised learning, it is easy to extend it to tackle supervised learning problems.
- Recall that VQ seeks to find areas of high density in high dimensional data by strategically placing codewords (in an online or batch approach).

Consider the online version of LVQ.

1. For each of the K classes, initialise R prototypes (representative points) to model each class distribution.
2. Sample an observation X and let V_c be the Voronoi region where it falls with cluster center μ_c.
3. If the prototype is of the same class as X, move μ_c towards X

$$mu_c \leftarrow \mu_c + \alpha(t)\left[X - \mu_c\right]$$

and if μ_c is of a different class, move it away from X

$$\mu_c \leftarrow \mu_c - \alpha(t)\left[X - \mu_c\right]$$

Repeat 2-3 many times and return the codebook.

Nearest Neighbours in High Dimensions

We have seen various ways to find nearest neighbors and the corresponding classification is intuitive in 2, 3 and general low-dimensional problems. The concept of a nearest neighbour is questionable, however, in high dimensions. First, look at multi-variate normal data in p dimensions,

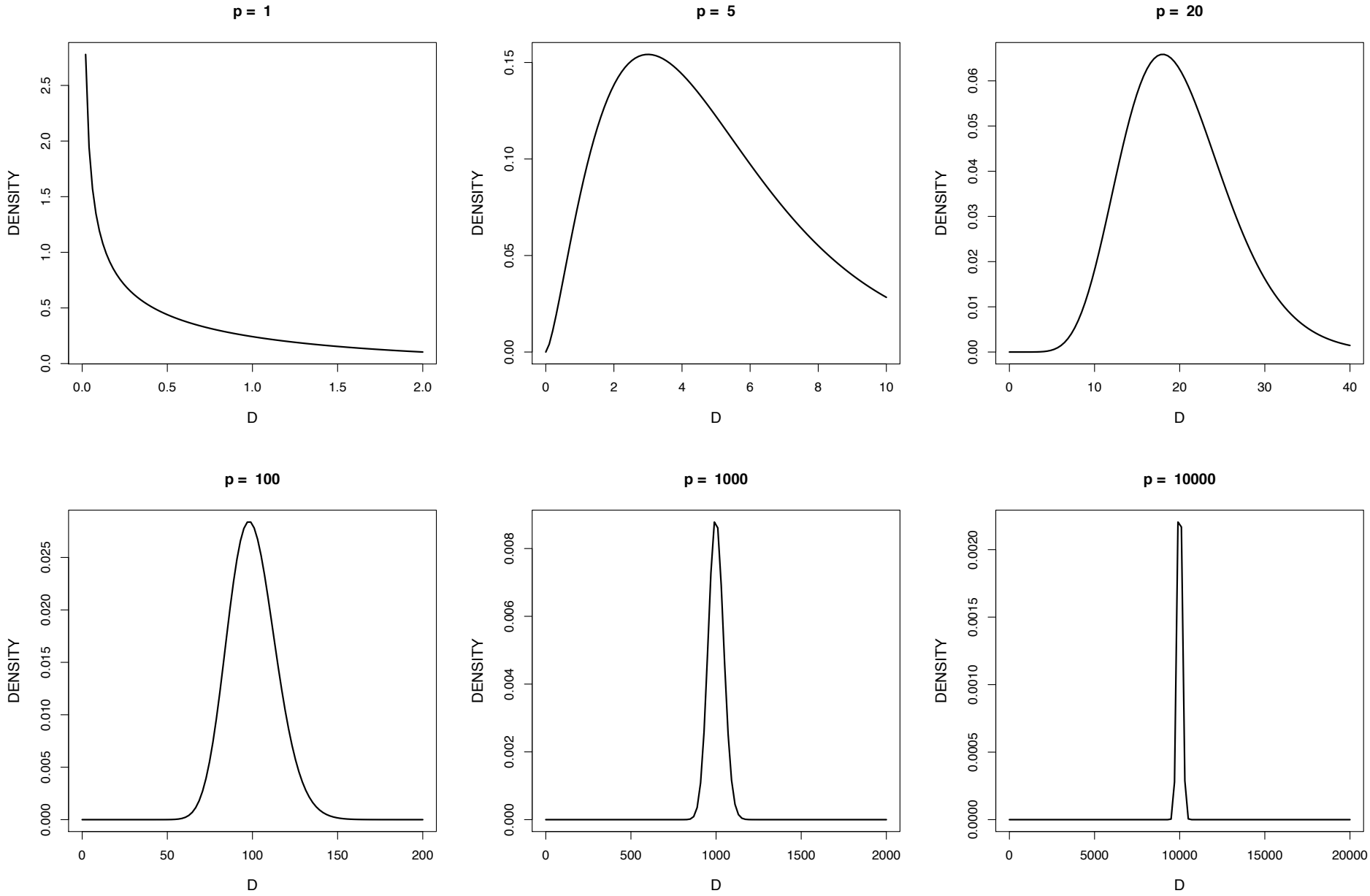
$$X \sim \mathcal{N}(\mu, \Sigma).$$

What is the distribution of the Euclidean distance D between a random observation X and the 'cluster center' μ if $\Sigma = 1_p$? It is

$$D = \sum_{k=1}^{p} (X^{(k)} - \mu^{(k)})^2.$$

And D has thus a χ^2_p-distribution with p degrees of freedom.

Density of distance D of observation from cluster center in p dimensions.



kNN in High Dimensional Spaces

Assume you have $\{X_i\}_{i=1}^n$ in $\mathbb{R}^p$ where $X_i \overset{\text{i.i.d.}}{\sim} f$.
Proposition. If we have

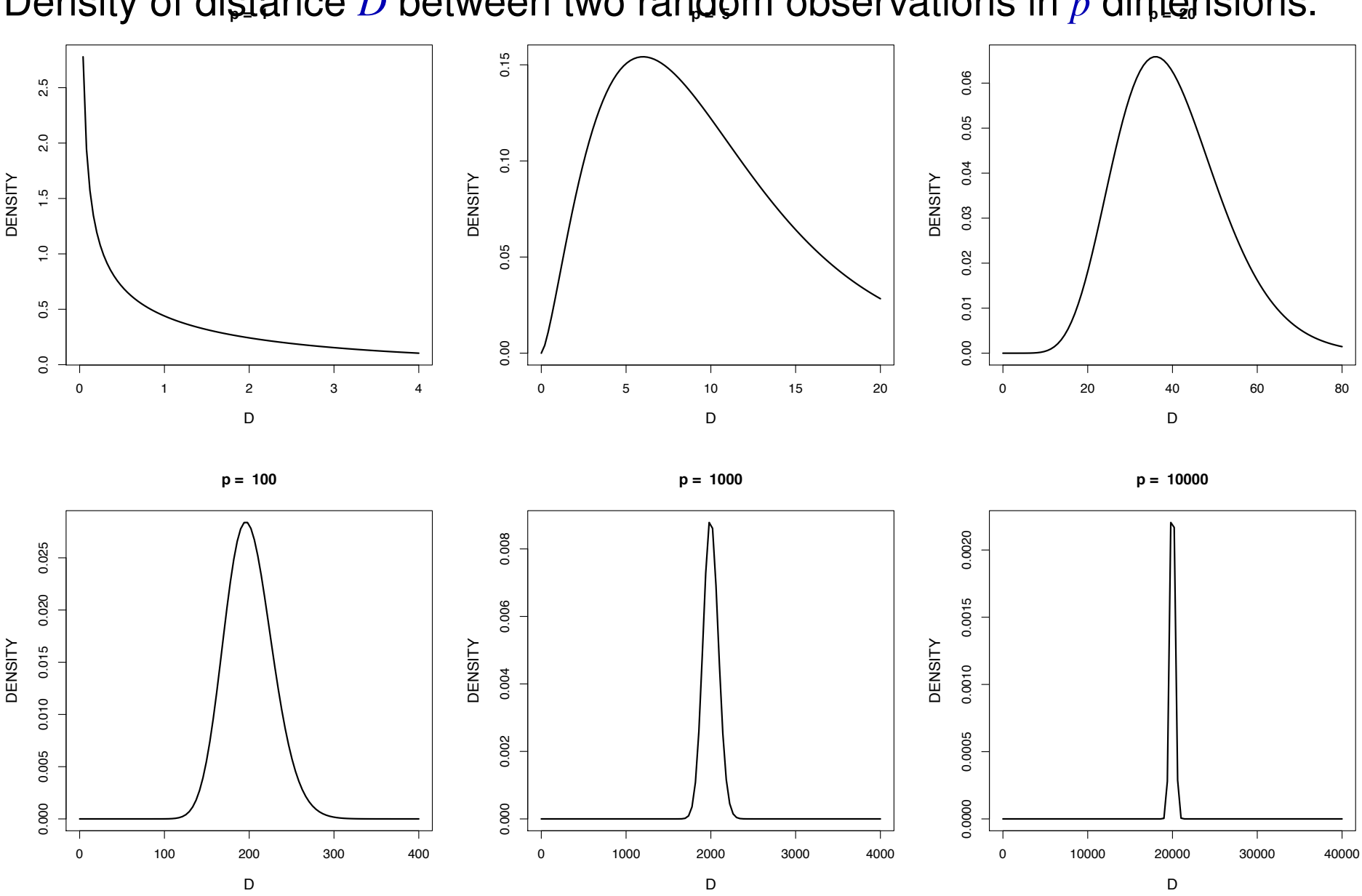
$$\lim_{p\to\infty} \frac{\mathbb{V}_f\left[d\left(X,x\right)\right]}{\mathbb{E}_f\left[d\left(X,x\right)\right]^2} = 0$$

then for any $\varepsilon > 0$

$$\lim_{p\to\infty} \mathbb{P}_{f^{\otimes n}}\left(\left|\max_{1\leq i\leq n} d\left(X_i,x\right) - \min_{1\leq i\leq n} d\left(X_i,x\right)\right| \geq \varepsilon\right) = 0.$$

Loosely speaking, in high dimensional spaces, all the points are at the same distance from the query point x so kNN is useless.
Example: Assume $d\left(X,x\right) = \sum_{l=1}^p \left(X^l - x^l\right)^2$ where $x^l = (\mu, ..., \mu)$ and $f\left(x\right) = \prod_{l=1}^p \mathcal{N}\left(x^l; 0, 1\right)$ then $d\left(X,x\right)$ follows a non-central chi-squared of variance $2p\left(1+2\mu^2\right)$ and mean $p\left(1+\mu\right)$ so that $\lim_{p\to\infty} \frac{\mathbb{V}_f[d(X,x)]}{\mathbb{E}_f[d(X,x)]^2} = 0$..

Density of distance $\tilde{D}$ between two random observations in p dimensions.



There are no real 'nearest neighbours' in high dimensions. All points are about the same distance from each other and are sitting on the shell of the high-dimensional sphere.

Now suppose there are two groups μ_1 and μ_2, where for the two classes the distributions of $X = (X^{(1)}, \ldots, X^{(p)})$ are, respectively,

$$\mathcal{N}(\mu_1, \Sigma) \qquad \text{and} \qquad \mathcal{N}(\mu_2, \Sigma),$$

with

$$\mu_1 = (2, 0, 0, 0, \ldots, 0)^T \qquad \text{and} \qquad \mu_2 = (0, 0, 0, 0, \ldots, 0)^T,$$

and $\Sigma = 1_p$. The two groups distinguish themselves thus just in the first component $X^{(1)}$.
Suppose we have n observations $X_1, \ldots, X_{2n}$, of which n are in class 1 and n in class 2. What is the probability $P(\text{correct classification})$ that a randomly chosen X from class 1 will have a nearest neighbor in $\{i : Y_i = 1\}$ rather than in $\{i : Y_i = 2\}$?

$$P(\text{correct classification}) = P(\min_{i:Y_i=1} \|X - X_i\|_2 \leq \min_{i:Y_i=2} \|X - X_i\|_2).$$

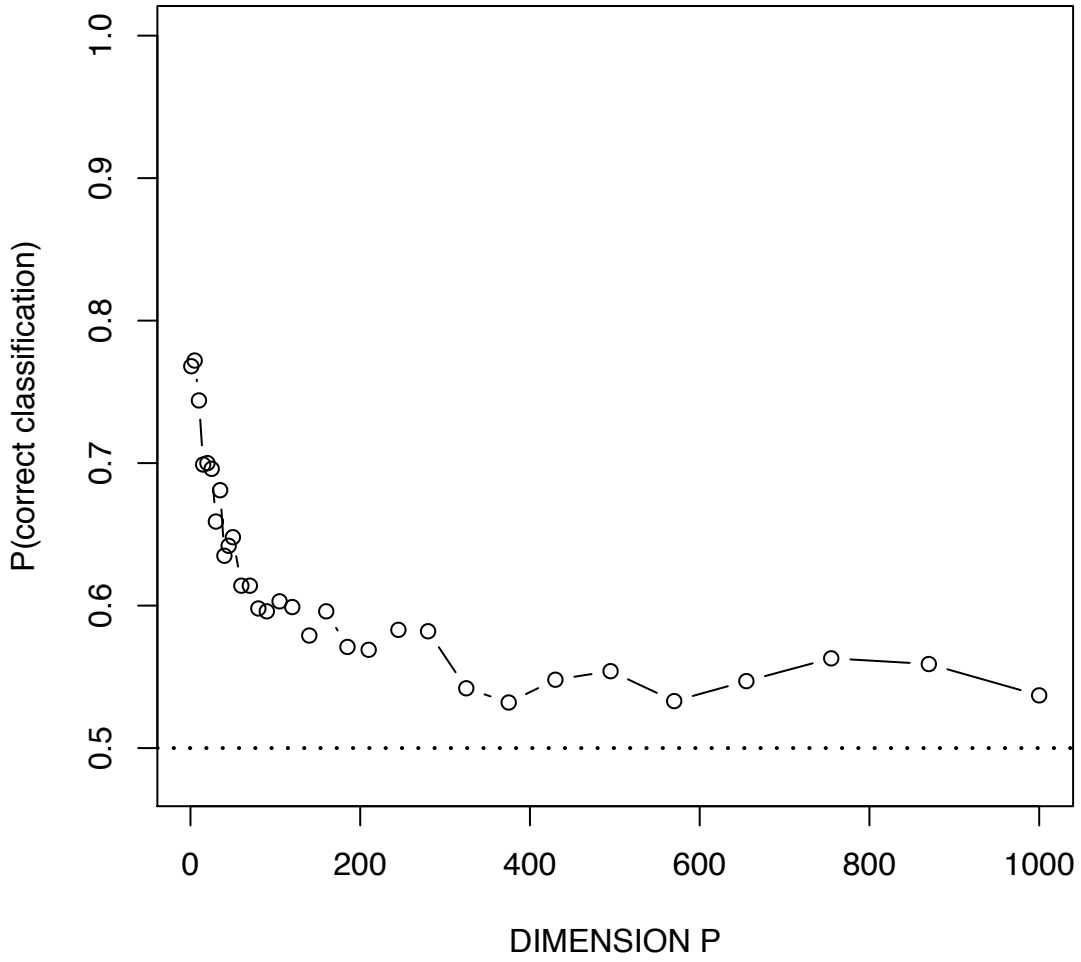
Answer easiest by simulation...

```
pvec <- pmax(1,unique(round((1/5)*exp(seq(0,log(1000),length=50)))*5))
nsim <- 1000
n <- 100
probability <- rep(0,length(pvec))
for (pc in 1:length(pvec)){
  p <- pvec[pc]
  for (sim in 1:nsim){

    X1 <- matrix(rnorm(n*p),nrow=n)
    X2 <- matrix(rnorm(n*p),nrow=n)
    X2[,1] <- X2[,1] + 2
    X <- rnorm(p)

    distance1 <- numeric(n)
    distance2 <- numeric(n)
    for (k in 1:n){
      distance1[k] <- mean( (X-X1[k,])^2 )
      distance2[k] <- mean( (X-X2[k,])^2 )
    }
    winningclass1 <- min(distance1)<min(distance2)
    if(winningclass1)  probability[pc] <- probability[pc] + 1/nsim
  }
  plot(pvec,probability,
    xlab="DIMENSION P",ylab="P(correct classification)",type="b")
}
```

Probability of correct classification with nearest neighbours as a function of dimension p. Misclassification probability of 0.5 can be achieved by random guessing (dotted line).



Nearest neighbor potentially performs poorly in high dimensions.