Lecture 7: The Curse of Dimensionality

Iain Styles

25 October 2019

The Curse of Dimensionality

Why might the dimensionality of the data affect the classification accuracy of a nearest-neighbours approach? We will spend this lecture trying to understand why high dimensionality is different. But first, we will try something ridiculous to see what difference the dimensionality makes. We will take each of our image vectors, which have $28 \times 28 = 784$ elements, and we will take their scalar (dot) product with each of 40 784-dimensional *random* vectors; vectors where the components are drawn independently from a normal distribution $\mathcal{N}(0,1)$ with mean 0 and variance 1. For image vectors $\mathbf{v}_i$, and random vectors $\mathbf{r}_i$ (both assumed to be column vectors) we compute new vectors

$$\begin{pmatrix} \mathbf{z}_1 & \mathbf{z}_2 & \dots & \mathbf{z}_N \end{pmatrix} = \begin{pmatrix} \mathbf{r}_1^{\mathrm{T}} \\ \mathbf{r}_2^{\mathrm{T}} \\ \dots \\ \mathbf{r}_M^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{v}_N \end{pmatrix}. \tag{1}$$

The $\mathbf{z}_i$ are said to be *random projections* of the original image vectors $\mathbf{v}$. There is one of these per sample in the dataset, but each now has only 40 components, instead of the original 768. Superficially this seems to be a very strange thing to do indeed: why would we even consider this? That is a question to which we will return in due course. For the moment though, let us just try it. We form new training and test sets by randomly projecting every element of both sets onto the same 40 random vector, and we apply $k$nn, with $k = 7$ to the random projections. The results of this are shown in Figure 1.

Something quite dramatic has happened. The overall accuracy has increased from 75% to 87%! The troublesome characters 3, 4 and 5 are now classified with $> 80\%$ accuracy. The random projection must have done something quite dramatic to our data. We will next try to understand what has happened, and why. It turns out that computing distances between high-dimensional vectors is not as innocuous as it might seem, and there is a hidden danger.

The Curse of Dimensionality

The underlying reason that reducing the dimensionality of the data using a random projection was useful is that the properties of high dimensional vector spaces are strange and counter intuitive. For a simple example of how our intuition breaks down, consider the following example.

T \ P	0	1	2	3	4	5	6	7	8	9
0	98	0	0	0	0	1	0	0	1	0
1	0	100	0	0	0	0	0	0	0	0
2	3	4	79	1	0	1	4	3	5	0
3	0	4	2	84	0	1	1	3	2	3
4	0	1	0	0	85	0	1	2	2	9
5	0	2	0	3	1	86	4	2	1	1
6	1	0	0	0	1	6	91	1	0	0
7	0	3	1	1	1	0	0	91	0	3
8	1	0	5	13	3	4	1	2	70	1
9	0	1	0	0	6	0	0	2	2	89

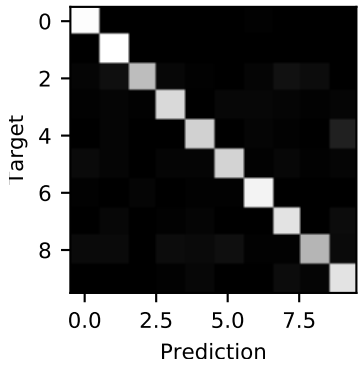

Figure 1: MNIST classification results (Target T vs Prediction P) with kNN for $k = 7$ using 40 random projections

A *hypercube* is the n-dimensional analogue of a square in 2d, and cube in 3d. In each dimension, the cube has a side of length $2r$ so that the centre of each of its faces is a distance r from the centre of the hypercube. Consider now the *hypersphere* that the hypercube encloses. The hypersphere, which is defined as the set of points a distance r from some central point, intersects with the hypercube only at the centres of the faces. This is shown in 3d is Figure 2

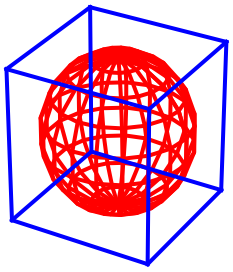

Figure 2: A sphere in 3d and its enclosing hypercube.

By definition, all of the points on the sphere are a distance r from the centre. The faces of the cube are also r from the centre. How far away from the centre are the corners of the hypercube? The answer is trivial to compute, but has surprising implications. In 2d, the corner of a square is $\sqrt{r^2 + r^2} = r\sqrt{2}$ from the centre. In 3d, the corner of a cube is $\sqrt{r^2 + r^2 + r^2} = r\sqrt{3}$. The pattern is obvious and follows directly from Pythagoras' theorem. In n-dimensions, the corner of a hypercube (of side length $2r$) is $\sqrt{\sum_{i=1}^{n} r^2} = r\sqrt{n}$. We plot this relationship in Figure 3. Remembering that the surface of the hypersphere is always r from the origin, we see that the distance to the corner is many times the sphere radius in high dimensions. In other words, although the sphere touches the cube in the centre of all of its faces, it does not get near to the corners.

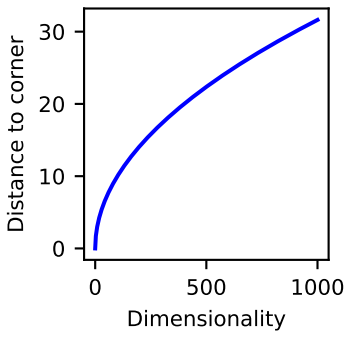

Figure 3: The distance from the centre to the corners of a hypercube of side $r = 1$ increases as the square-root of the dimensionality.

Let us look at this in a different way and consider the fraction of the the volume of the hypercube that is occupied by its enclosed hypersphere. In 2d, a square of side $2r$ has area $4r^2$. The enclosed

circle has area πr^2, and so the circle occupies a fraction $\pi/4 \approx 0.785$ of the square. In 3d, the cube has volume $8r^3$; the sphere has volume $4\pi r^3/3$, and the ratio is $4\pi/24 = 0.52$. This is already a substantial decrease! In n-dimensions, the volume of the hypercube is $(2r)^n$, and the volume of the hypersphere can be shown to be given by $\frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2}+1)}R^n$, where $\Gamma(x)$ is the Gamma function, an extension of the factorial function that is beyond the scope of this module to study in detail. Some information can (naturally!) be found at `https://en.wikipedia.org/wiki/Gamma_function`. Using these results, we plot the sphere/cube ratio in Figure 4.

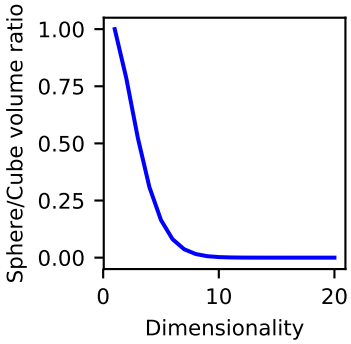

Figure 4: The ratio of the volume of a hypersphere to its enclosing hypercube as a function of dimensionality.

It is remarkable how quickly the ratio drops to zero: in 10d, nearly all of the volume in the cube is outside of the sphere. When combined with our previous result on the distance from the centre of the cube to its corners, we may conclude that nearly all of the volume of a hypercube is near to the corners, and that almost no volume is near the centre. This can, if you so wish, be verified numerically by constructing a grid of evenly spaced points in high dimensions and seeing how many are within distance r of the centre: very few.

To conclude this discussion, consider two hyperspheres, of radius r and $r-\delta$ respectively (with $\delta \ll r$). The volume of the larger sphere is αr^n, where α is a constant that depends on the dimensionality n; and the volume of the smaller sphere is $\alpha(r-\delta)^n$. The volume of the "shell" that is inside the larger sphere but outside the smaller sphere is therefore $\alpha\left(r^n - (r-\delta)^n\right)$. As a proportion of the larger sphere, the shell occupies a volume

$$\frac{V_{\text{shell}}}{V_{\text{sphere}}} = \frac{\alpha\left(r^n - (r-\delta)^n\right)}{\alpha r^n} \tag{2}$$

$$= 1 - r^{-n}(r-\delta)^n \tag{3}$$

$$= 1 - \left(r^{-1}(r-\delta)\right)^n \tag{4}$$

$$= 1 - \left(1 - \frac{\delta}{r}\right)^n \tag{5}$$

Taking the limit as the dimension tends to infinity gives

$$\lim_{n\to\infty} \frac{V_{\text{shell}}}{V_{\text{sphere}}} = \lim_{n\to\infty} 1 - \left(1 - \frac{\delta}{r}\right)^n \tag{6}$$

$$= 1 \tag{7}$$

because $1 - \frac{\delta}{r} < 1$. Thus most of the volume of the hypersphere is concentrated in a thin shell around its edge and most of the volume is at its edge.

Although these results are intrinsically interesting, it is not clear that they are immediately relevant to our problem of doing distance-based classification of MNIST digits, so let us perform a simple numerical experiment. For a range of dimensionalities, we generate 10^6 uniformly randomly distributed data points and compute the distances between all pairs of points. We then find

the maximum and minimum distances between points and compute the ratio of the range of distances as compare to the minimum distance, $(d_{\max} - d_{\min})/d_{\min}$. The resulting graphs are shown in Figure 5.

The figure provides an empirical verification of a well-known result:

$$\lim_{n\to\infty} \mathbb{E}\left(\frac{d_{\max} - d_{\min}}{d_{\min}}\right) \to 0 \tag{8}$$

which is a statement that in high dimensions, the difference between minimum and maximum distances between points tends towards zero, and hence all distances become "similar". This result has profound implications for any algorithm that requires computation and comparison of pairwise distances in high dimensions, and is commonly referred to as the *curse of dimensionality*.

To what extent is this relevant in MNIST? To test this, we compute the pairwise distances between 1000 points from the test set and 1000 points from the training set and plot a histogram of their distribution (Figure 6). This shows that there are no "small" distances in the data, a direct result of each pairwise distance being the sum of the squares of each of 784 components, and the distribution of distances is roughly normal, with a mean/median of ≈ 2300 and a standard deviation of ≈ 300. This means that 68% of pairwise distances lie between 2000 and 2600, and 95% between 1700 and 2900. The range of distance is indeed compressed, but perhaps not as much as we might have expected. Why is this?

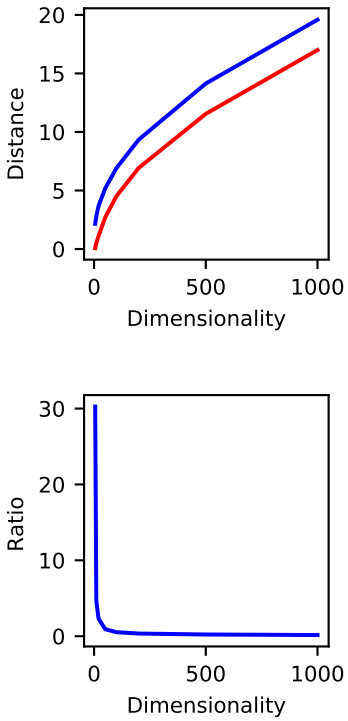

Figure 5: *Top*: Minimum (red) and maximum (blue) pairwise distances in a set of 10^6 uniformly randomly distributed data points as a function of dimensionality. *Bottom*: The ratio $(d_{\max} - d_{\min})/d_{\min}$.

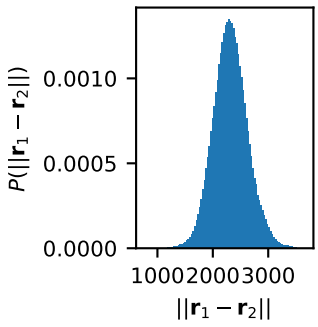

Figure 6: Distribution of pairwise distances in pixel space between 1000 examples from the MNIST test set, and 1000 examples from the training set.

Reading

Section 1.4 of Bishop, Pattern Recognition and Machine Learning has a good discussion of the curse of dimensionality. An expansive treatment on the role that randomised methods can serve in high-dimensional problems is given in the article “Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions”, N. Halko, P. G. Martinsson, J. A. Tropp. SIAM Review Vol. 53, No. 2, pp. 217âĂŞ288 (2011). This is one of my favourite articles ever, and has been very influential in my own work.