Ex.2.1 Suppose each of K-classes has an associated target t_k, which is a vector of all zeros, except a one in the kth position. Show that classifying to the largest element of $\hat{y}$ amounts to choosing the closest target, $\min_k ||t_k - \hat{y}||$, if the elements of $\hat{y}$ sum to one.

Solution:

$$\begin{aligned} ||t_k - \hat{y}|| &= \sqrt{\hat{y}_1^2 + \hat{y}_2^2 + \cdots + (\hat{y}_k - 1)^2 + \ldots + \hat{y}_p^2} \\ &= \sqrt{\hat{y}_1^2 + \cdots + \hat{y}_p^2 - 2\hat{y}_k + 1} \end{aligned}$$

All terms other than $-2\hat{y}_k$ are independent of k. Since square root monotonically increases, the lesser the $-2\hat{y}_k$ term is, the lesser the whole expression. This is equivalent to maximizing $\hat{y}_k$, which in turn corresponds to picking $\hat{y}_k$ to be the largest element of $\hat{y}$.

Ex.2.3 Consider a p-dimensional unit ball centered at the origin as our input space. Suppose that our data consists of N points uniformly distributed in this region. What is the median distance from the origin to the nearest point?

Solution:

Since we are in a unit ball, our distance ranges from 0 to 1. By definition, the probability of the nearest point being further or nearer than our median c is 0.5. So, we solve $P(d_{min} < c) = 0.5$. To find $P(d_{min} < c)$, we first find $P(d < c)$ for single point, where d is distance. This is simply $\frac{V(ball_c)}{V(ball_1)}$, since $ball_c$ contains all points with d less than c. Correspondingly, we have $P(d > c) = 1 - \frac{V(ball_c)}{V(ball_1)} = 1 - c^p$ since the volume of a ball scales with its radius raised to the p-th power.

Since points are independent, $P(d_{min} < c) = 1 - P(d > c)^N = 1 - (1 - c^p)^N$, from our result above. Now we set $1 - (1 - c^p)^N = 0.5$ and solve:

$$\begin{aligned} \tfrac{1}{2} - (1 - c^p)^N &= 0 \\ \tfrac{1}{2}^{\frac{1}{N}} &= 1 - c^p \\ c &= (1 - \tfrac{1}{2}^{\frac{1}{N}})^{1/p} \end{aligned}$$

We can analyze this formula to see what it means. The value inside parentheses is bounded by 0 and 1/2, and it decreases from 1/2 when $N = 1$ to 0 as $N \to \infty$. This makes sense, since the more points we have, the less sparse the data and thus the closer the nearest point should be. However, this value is raised to the $\frac{1}{p}$ power, which means that c increases greatly as p

increases. So the more points, the less sparse, but the more dimensions, the more sparse. Let's see what happens when both N and p grow:

$$\begin{aligned}
L &= \lim_{N,p\to\infty} c \\
\ln L &= \lim_{N,p\to\infty} \ln\left(\left(1-(1/2)^{\frac{1}{N}}\right)^{\frac{1}{p}}\right) \\
&= \lim_{N,p\to\infty} \frac{1}{p}\ln\left(1-(1/2)^{\frac{1}{N}}\right) \\
&= \lim_{N,p\to\infty} \frac{1}{p}\ln\frac{2^{\frac{1}{N}}-1}{2^{\frac{1}{N}}} \\
&= \lim_{N,p\to\infty} \frac{1}{p}\ln 2^{\frac{1}{N}}-1 \\
&= \lim_{N,p\to\infty} \frac{1}{p}\ln 2^{\frac{1}{N}} \\
&= \lim_{N,p\to\infty} \frac{1}{Np}\ln 2 \\
&= 0 \\
\implies L &= 1
\end{aligned}$$

So, asymptotically, the median distance to the nearest point approaches 1, which puts the point near the boundary. A large enough p will cause points to accumulate near the boundary of the input space and the effect cannot be nullified with more data points. This effect holds even with a different input space and non-uniform distributions (such as a Normal model). This causes problems for function approximation models based off of local neighborhoods, such as K-nearest neighbors, since they rely on training data near the input. This is known as the *curse of dimensionality*, discussed further in *ESL* section 2.5.

Ex.2.4 The edge effect problem discussed on page 23 is not peculiar to uniform sampling from bounded domains. Consider inputs drawn from a spherical multinormal distribution $X \sim \mathcal{N}(0, \mathbf{I}_p)$. The squared distance from any sample point to the origin has a χ^2_p distribution with mean p. Consider a prediction point x_0 drawn from this distribution, and let $a = x_0/||x_0||$ be an associated unit vector. Let $z_i = a^T x_i$ be the projection of each of the training points on this direction.

Show that the z_i are distributed $\mathcal{N}(0, 1)$ with expected squared distance

from the origin 1, while the target point has expected squared distance p from the origin.

Solution:

Let $a = (a_0, a_1, \ldots, a_p)$. Then, $a^T X = a_0 x_0 + a_1 x_1 + \cdots + a_p x_p$. Since all elements of X are independent (the covariance matrix is identity), $x_i \sim \mathcal{N}(0,1)$ for all i. Since z_i is a sum of scaled standard normal random variables, z_i is also normal. Furthermore, its mean is:

$$a_0\mu(x_0) + a_1\mu(x_1) + \cdots + a_p\mu(x_p) = 0.$$

Its variance is:

$$a_0^2\mathrm{Var}(x_0) + a_1^2\mathrm{Var}(x_1) + \cdots + a_p^2\mathrm{Var}(x_p) = a_0^2 + a_1^2 + \cdots + a_p^2 = 1$$

since a is a unit vector. Therefore, $z_i \sim \mathcal{N}(0,1)$. Expected squared distance is:

$$\mathrm{E}\left[z_i^2\right] = 1$$

since chi-squared has expectation equal to its degrees of freedom.

Ex.2.6 Consider a regression problem with inputs x_i and outputs y_i, and a parameterized model $f_\theta(x)$ to be fit by least squares. Show that if there are observations with tied or identical values of x, then the fit can be obtained from a reduced weighted least squares problem.

Solution:

The residual squared sum (RSS), is

$$\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{N}(y_i - f_\theta(x_i))^2.$$

Since this is a non-negative quadratic in θ, then we can take its derivative and set it to zero to obtain the global minimum. Since we are taking a vector derivative, the derivative with respect to every parameter θ_k must be zero. So we have:

$$\frac{\partial \mathrm{RSS}}{\partial \theta_k} = \sum_{i=1}^{N} 2(y_i - f_\theta(x_i))\frac{\partial f_\theta(x_i)}{\partial \theta_k} = 0.$$

Suppose we have to inputs, x_α and x_β, that are identical (tied). In that case, we can re-write $\frac{\partial \text{RSS}}{\partial \theta_k}$ (let's call it RSS'_k) as:

$$\begin{aligned}
\text{RSS}'_k &= \sum_{i=1, i\neq\alpha,\beta}^{N} 2(y_i - f_\theta(x_i))\frac{\partial f_\theta(x_i)}{\partial \theta_k} + 2(y_\alpha - f_\theta(x_\alpha))\frac{\partial f_\theta(x_\alpha)}{\partial \theta_k} + 2(y_\beta - f_\theta(x_\beta))\frac{\partial f_\theta(x_\beta)}{\partial \theta_k} \\
&= \sum_{i=1, i\neq\alpha,\beta}^{N} 2(y_i - f_\theta(x_i))\frac{\partial f_\theta(x_i)}{\partial \theta_k} + 2(y_\alpha + y_\beta - 2f\theta(x_\alpha))\frac{\partial f_\theta(x_\alpha)}{\partial \theta_k} \\
&= \sum_{i=1, i\neq\alpha,\beta}^{N} 2(y_i - f_\theta(x_i))\frac{\partial f_\theta(x_i)}{\partial \theta_k} + 4(\text{Avg}(y_\alpha, y_\beta) - f\theta(x_\alpha))\frac{\partial f_\theta(x_\alpha)}{\partial \theta_k}
\end{aligned}$$

Here, we notice that the last term looks exactly like the other terms being summed, only twice as large and with two outputs average instead of one output. In fact, this is where the reduction happens. We have effectively removed one of the inputs and combined it into one term where we only reference x_α. Finding the antiderivative of this, we find our new RSS, its derivative is identitical, thus lending us the same fit θ. If we have more than two copies of any input, then we simply pick two, combine them, re-write the RSS, and repeat until we have no repeated inputs. The more general formula for this, for c repetitions of an input, is:

$$\sum_{unique} 2(y_i - f_\theta(x_i))\frac{\partial f_\theta(x_i)}{\partial \theta_k} + 2c(\text{Avg}(y_{repeats}) - f\theta(x_\alpha))\frac{\partial f_\theta(x_\alpha)}{\partial \theta_k}$$

Where x_α is the repeated input value. We find that an appropriate anti-derivative is:

$$\text{RSS}(\theta) = \sum_{unique;i} (y_i - f_\theta(x_i))^2 + \sum_{repeats;j} c_j(\text{Avg}(y_j) - f_\theta(x_j))^2$$

Here, $j \in J$ is an element of an index set indexing all of the repeated values. y_j is the set of all outputs corresponding to the jth repeated value. c_j is the total quantity of copies, and x_j is the input value itself. This is a different sum-of-squares, but since its derivative is the same as our initial problem, then it will provide the same fitting function $f_\theta(x)$. This will help us for problems with lots of tied inputs by reducing the size of the input matrix $\mathbf{X}$ and the output vector $\mathbf{y}$. Since matrix operations (specifically inversion) is expensive, this can save us a lot of computing time.