Homework 2: The Curse of Dimensionality

DS-GA 1008 Deep Learning

Fall 2021

Homework 2 is to let you better understand the curse of dimensionality in theory and practice.

For theory questions, you should put all your answers in a PDF file and we will not accept any scanned hand-written answers. It is recommended to use LaTeX.

For coding questions (Q2, Q12 and Q14), you need to program in Python. We recommand you to use **PyTorch** and **NumPy** for the experiments and **Matplotlib** to draw the plot. You need to submit your python files and describe the results with plots in the pdf.

If you only submit your python files without relevant description and plots in the pdf, you will get no points for the questions.

In homework 2, the questions have 25 points in total. We set 5 bonus points and it means that **you will have full credits if you get 20 points** and extra credits for more points.

The due date of homework 2 is `23:55 EST` of `10/11`. Submit the following files in a zip file `your_net_id.zip` through NYU Brightspace:

- `HW2.pdf`
- `q2.py`, `q12.py` (optional) and `q14.py`

The following behaviors will result in penalty of your final score:

1. 5% penalty for submitting your files without using the correct format. (including naming the zip file, PDF file or python file wrong, or adding extra files in the zip folder, like the testing scripts from coding questions).
2. 20% penalty for late submission within the first 24 hours. We will not accept any late submission after the first 24 hours.
3. 20% penalty for code submission that cannot be executed using the steps we mentioned in coding questions. So please test your code before submit it.

Problem 1: The Curse of Dimensionality

In this problem, we will study a high-dimensional statistical phenomena called the *Curse of Dimensionality*. Originally coined by Bellman in the context of dynamic programming, it generally refers to the failure of certain statistical or algorithmic procedures to scale efficiently as the input dimension grows.

We will first study this curse through the geometry of the unit d-dimensional ℓ_2 sphere. Recall that the ℓ_p norm of a vector $x \in \mathbb{R}^d$ is defined as $\|x\|_p := (\sum_i |x_i|^p)^{1/p}$, for $p \in [1, \infty)$. To get a first grasp on the phenomena, we will first consider a 'Gaussian' approximation of the d-dimensional unit sphere, by considering a Gaussian random vector $X \sim \mathcal{N}(0, I/d)$, where I is the $d \times d$ identity matrix.

1. Using the Central Limit Theorem, show that $\|X\|^2 = 1 + O(1/\sqrt{d})$ with high probability. In other words, show that $\|X\|^2$ is a random variable with expectation 1 and standard deviation proportional to $1/\sqrt{d}$. [Alternative: use the χ-squared distribution with d degrees of freedom]. (**1 point**)

2. Numerically verify this property by simulating $\|X\|^2$ in dimensions $d \in \{10, 100, 1000, 10000\}$ using a sample size of $n = 1000$. (**2 points**)

This means that for large input dimension d, a draw X from the $\mathcal{N}(0, I/d)$ Gaussian distribution will concentrate to the unit sphere $\|X\|_2 \approx 1$. Let us also verify that the Gaussian distribution is rotationally invariant:

3. Let $R \in \mathbb{R}^{d \times d}$ be any unitary matrix, ie $R^\top R = R R^\top = I$. Show that the pdf of $X \sim \mathcal{N}(0, I/d)$ and $\tilde{X} = RX$ are the same, ie they do not depend on the choice of R. (**2 points**)

For our purposes, we will thus use Gaussian $\mathcal{N}(0, I/d)$ samples as *de facto* draws from the unit d-dimensional sphere.

4. If we draw two datapoints X, X' i.i.d. from $\mathcal{N}(0, I/d)$, show that $|\langle X, X' \rangle| = O(1/\sqrt{d})$ with high probability using again the CLT. In other words, show that $\langle X, X' \rangle$ is a random variable of zero mean and standard deviation proportional to $1/\sqrt{d}$. Conclude that for a constant $C > 0$, $\|X - X'\| \in (\sqrt{2} \pm C/\sqrt{d})$ for large d with high probability. (**2 points**)

This property reflects the intuition that independent draws from a high-dimensional Gaussian distribution (or any rotationally-invariant distribution, more generally) are nearly orthogonal as $d \to \infty$, since $|\langle X, X' \rangle|$ will have variance going to 0 as $d \to \infty$.

We will now combine this intuition with a simple supervised learning setup. Assume a target function $f^* : \mathbb{R}^d \to \mathbb{R}$ is β-*Lipschitz*, ie $|f^*(x) - f^*(x')| \leqslant \beta \|x - x'\|$, and a dataset $\{(x_i, y_i)\}_{i \dots n}$ with $x_i \sim \mathcal{N}(0, I/d)$ and $y_i = f^*(x_i)$ drawn independently. We will consider the *Nearest-Neighbor* estimator $\hat{f}_{\mathrm{NN}}$ given by

$$\hat{f}_{\mathrm{NN}}(x) := y_{i(x)} \text{ , where } i(x) = \arg\min_i \|x - x_i\| \text{ .}$$

5. Show that $\mathbb{E}|\hat{f}_{\mathrm{NN}}(x) - f^*(x)| \leqslant \beta \mathbb{E} \min_i \|x - x_i\|$, where the expectation is taken over the test sample x and the training sample $\{x_i\}$. (**2 points**)

6. Let E_n denote the expectation of $\min_{i=1\dots n} Y_i$, where $Y_i \sim \mathcal{N}(0,1)$ i.i.d. Show that if now $X_i \sim \mathcal{N}(\mu,\sigma^2)$ i.i.d., then $\mathbb{E}\min_i X_i = \mu + \sigma E_n$. (**1 point**)

7. Using the fact that $\|x-x_i\|$ and $\|x-x_j\|$ are conditionally independent, given x, and assuming the asymptotic Gaussianity of $\|X-X'\| \xrightarrow{d} \mathcal{N}(\sqrt{2}, C/d)$ for a constant $C>0$, show that $\mathbb{E}\min_i \|x-x_i\| \sim \sqrt{2} + \frac{\sqrt{C}}{\sqrt{d}} E_n$. (**1 point**)

8. Using the fact that $E_n \approx -\sqrt{2\log n}$, conclude that as long as $\log n \ll d$, the generalisation bound in (5) is vacuous (in other words, unless the sample size is exponential in dimension, the bound does not provide any useful learning guarantee). (**2 points**)

While this argument shows that our Lipschitz-based upper bound is cursed by dimension, let us conclude this exercise by showing that the exponential dependency in dimension is necessary. For simplicity, we will replace the d-dimensional ℓ_2 sphere by the ℓ_∞ sphere, ie the cube $\mathcal{B} = [-1,1]^d$. Let $\Omega = [-1/2, 1/2]^d$ denote a smaller cube. For $x \in \Omega$, let $\Psi(x) = \mathrm{dist}(x, \partial\Omega)$. In words, $\Psi(x)$ is the distance from x to the boundary of the cube Ω.

9. Show that Ψ is 1-Lipschitz. [*Hint*: Use the triangular inequality, and the definition of Ψ as $\Psi(x) = \min_{y\in\partial\Omega} \|x-y\|$. Also, drawing Ψ in two-dimensions might help!] (**2 points**)

We will now use this 1-Lipschitz function supported in the 'small' cube Ω to construct a hard-to-learn function f^* defined in the large cube $\mathcal{B} = [-1,1]^d$.

10. Verify that we can fit 2^d copies of Ω into $\mathcal{B}$, by translating copies of Ω appropriately [*Hint:* Use the fact that both Ω and $\mathcal{B}$ are separable in the standard basis. Also, drawing this in two-dimensions should help]. (**2 points**)

Let now $g \in \{\pm 1\}^{2^d}$ be a binary string of length 2^d, that we index using d binary variables $z_1 = \pm 1, \dots z_d = \pm 1$. We define

$$f^*(x) = \sum_{z=z_1=\pm 1,\dots z_d=\pm 1} g(z)\Psi(x - z/2)\ . \tag{0.1}$$

In words, f^* is constructed by tiling 2^d shifted versions of the window Ψ, and flipping the sign of each tile with the bit $g(z)$. From part 9, the support of f^* is the 'large' cube $\mathcal{B}$.

11. Verify that f^* is 1-Lipschitz. [*Hint:* Given two points $x, x' \in \mathcal{B}$, consider the line segment joining them, and let y_k be the intersections with boundaries of tiled Ω, treating each resulting segment separately.] (**1 point**)

12. For $d=2$, draw an instance of f^* (You can use either manual drawing or a drawing software). (**2 points**)

13. Finally, show that if n, the number of training samples, satisfies $n \leqslant 2^{d-1}$, then the generalisation error of *any* learning algorithm producing $\hat{f}$ will be such that

$$\frac{\mathbb{E}_{x\sim \mathrm{Unif}([-1,1]^d)} |f^*(x) - \hat{f}(x)|}{\mathbb{E}_{x\sim \mathrm{Unif}([-1,1]^d)} |f^*(x)|} \geqslant 1/2\ .$$

In words, the relative generalisation error won't go to zero unless $n \gtrsim 2^d$. [*Hint:* Argue in terms of the tiling you have constructed in question 9; what happens if no datapoint intersects a given tile?] (**2 points**)

14. Choosing $d \in [5, 13]$, implement this experiment, using any predictor for f^* you want (e.g. a Neural Net), and the Mean-Squared Error (MSE) loss. Verify that the required sample size n before your model starts generalising grows with d exponentially. For each d, draw two large datasets $\{x_i\}_{i=1\ldots n}$, $\{\tilde{x}_i\}_{i=1\ldots n}$ with $x_i, \tilde{x}_i \sim \text{Unif}([-1,1]^d)$, then draw $K = 10$ different target functions f_k^*, $k = 1 \ldots K$ by picking random bits in equation (0.1), and fit your model to the training data $\{(x_i, y_i = f_k^*(x_i)\}_{i=1\ldots n}$. Then estimate your relative generalisation error using the test set $\{(\tilde{x}_i, \tilde{y}_i = f_k^*(\tilde{x}_i)\}$ (MSE error divided by standard deviation of the target function on test set), and average the performance across the K runs. You can pick $n \in \{2^j; j = 5 \ldots 16\}$. (**3 points**)