Modern Algorithms in Machine Learning: Exercises for Tutorials 3 & 4

Question 1. *What is the running time for applying the Random Projection Method to a set of P input vectors?*

Question 2. *(a bit tricky.) Demonstrate that it is essential for the Random Projection Method to choose the linear function f randomly. Specifically, prove that for any linear function $f : \mathbb{R}^N \to \mathbb{R}^M$ there are two vectors $x_1, x_2 \in \mathbb{R}^N$, $x_1 \neq x_2$, such that $f(x_1) = f(x_2)$.*
Hint: *Your argument should at some point exploit that $N > M$.*

Question 3. *(a bit tricky.) We now try to combine Dimensionality Reduction using the Random Projection Method with the k-NN algorithm. Assume that our test data consists of P vectors $x_1, x_2, \ldots, x_P \in \mathbb{R}^N$ with discrete labels $y_1, y_2, \ldots, y_P \in \{-1, +1\}$. Our training data consists of Q vectors $\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_Q \in \mathbb{R}^N$, which are unlabelled. How small do we need to choose ϵ so that for any point $\tilde{x}_j$, $1 \leq j \leq Q$, the set of the k nearest neighbours in the subspace of dimension M is equal to the set of k nearest neighbour in the original space of dimension N?*

Question 4. *Consider the deterministic weighted majority algorithm. Prove or disprove that the algorithm always makes a number of mistakes M^T within T rounds that satisfies $M^T \geq \min_{i \in [n]} m_i^T$.*

Question 5. *Consider online learning with n experts. Assume that each expert $i \in \{1, \ldots, n\}$ predicts each time wrongly with probability p_i, independently of the other experts and the time step. Design an algorithm that performs better than the weighted majority algorithm.*
Hint: *Use a maximum likelihood approach (a.k.a. MLE).*

Question 6. *Suppose there might be a few experts that make the wrong prediction in most steps – however, we do not know which these experts are in the beginning. How could you modify the Weighted Majority Algorithm (or the Randomised Weighted Majority Algorithm) so that it performs better in such cases?*

Question 7. *We have discussed in the lecture how we should pick the learning rate δ if T is known (or perhaps even $\min_{i \in [1,n]} m_i^{(T)}$), in order to optimise the bound on the number of mistakes. Can you suggest a way of adjusting δ with time t, if T is not known in advance?*
Hint: *Try to define $\delta(t)$ so that for any $T \geq 1$, $\sum_{t=1}^{T} \delta(t) \approx \sqrt{T \cdot \ln(n)}$.*

Question 8. *(a bit tricky.) Show that for any deterministic algorithm for online learning with n experts, there exists an input such that $M^T \geq 2 \cdot \min_{i \in [n]} m_i^T$.*
Hint: *Try to construct an input with two experts ($n = 2$).*