Foundations of Machine Learning – Homework assignment 1

CentraleSupélec / Université Paris-Saclay
Yannick Le Cacheux
yannick.le-cacheux@centralesupelec.fr

September 2024

As a reminder, this "homework" is entirely optional. As such, you may disregard the points awarded per exercise – they are used for another class.

I would advise you to try and do at least the first section "probability and statistics", as a reminder of tools that we will need for the class. You are of course very welcome to do the other sections when we have covered the relevant material in class.

I am currently looking for a method to send you the solutions in a way that does not make them publicly available on the Internet. As a reminder, please do not share these solutions publicly.

You are also welcome to send me your solution if you want some feedback.

1 Probability and statistics [25 points]

1.1 Bayes theorem [6 points]. A laboratory has a blood test for a disease which has a *sensitivity* of 0.95, which means that if a tested person has the disease, the test will have a 95% chance of being positive. The test also has a *false positive rate* of 1%, which means that if a tested person does not have the disease, the test will have a 1% risk of wrongly being positive. If the prevalence of the disease is 0.1%, i.e. the probability of a random person being infected is 1 out of 1000, what is the probability that a randomly tested person with a positive result is infected?

1.2 Maximum likelihood estimation. The normal distribution, also called the Gaussian distribution, is one of the most commonly encountered probability distributions. For given mean μ and standard deviation σ, its probability density function is

$$p(x|\mu,\sigma) = \mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right) \tag{1}$$

We assume that we have N independent and identically distributed (*i.i.d.*) samples $\mathbf{x} = \{x_1, x_2, \dots, x_N\}$ from this distribution. Recall that the *likelihood* of some data $\mathbf{x}$ given parameters $\boldsymbol{\theta}$ is given by $p(\mathbf{x}|\boldsymbol{\theta})$.

1.2.1 [5 points] Write the log-likelihood of the data, i.e. the logarithm of the likelihood. You may use a logarithm is base e, i.e. the natural logarithm ln.

In general, maximum likelihood is an estimation method for parameters $\boldsymbol{\theta}$ in which we select $\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\text{argmax}}\ p(\mathbf{x}|\boldsymbol{\theta})$.

1.2.2 [7 points] Setting the gradient of the log-likelihood with respect to μ to 0, derive the maximum likelihood estimator of μ.

1.2.3 [7 points] Using a similar method, derive the maximum likelihood estimator of σ^2.

1.2.a *Bonus question [+5 points]*: show that the maximum likelihood estimator of σ^2 is biased. An estimator $\hat{\boldsymbol{\theta}}$ is unbiased if $\mathbb{E}[\hat{\boldsymbol{\theta}}] = \boldsymbol{\theta}$.

2 Linear regression [35 points]

2.1. Parameters of a linear regression. In linear regression, for a given D-dimensional input variable $\mathbf{x} = (x_1\ x_2\ \ldots\ x_D)^\top \in \mathbb{R}^D$, we estimate a value y with $\hat{y} = \mathbf{w}^\top \mathbf{x} = \sum_{d=1}^{D} w_d x_d$. In the least square formulation, we assume that we are given N labeled observations $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, and we are looking for the parameters $\mathbf{w} \in \mathbb{R}^D$ which minimize

$$\frac{1}{N}\sum_{n=1}^{N}(y_n - \mathbf{w}^\top \mathbf{x}_n)^2 \tag{2}$$

We will assume here that input variables $\mathbf{x}_n$ are 2-dimensional, such that $\mathbf{w} = (w_1\ w_2)^\top = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$.

2.1.1 [8 points] Setting the partial derivatives of Equation (2) with respect to each parameter w_1, w_2 to 0, derive the value of $\mathbf{w}$ given the observations.

Recall from the lecture on linear regression that in general in D dimensions, noting $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^\top \in \mathbb{R}^{N \times D}$ our input variables and $\mathbf{y} = (y_1, \ldots, y_N) \in \mathbb{R}^N$ our labels, the parameters $\mathbf{w}$ can be obtained with

$$\mathbf{w} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} \tag{3}$$

2.1.2 [7 points] Are your results from Question 2.1.1 compatible with Equation (3)?

2.2 Least square loss. We will assume that our input variables have a single dimension. We will further assume that the labels $\{y_1, \ldots, y_N\}$ of our observations are given by $y = w \cdot x + \epsilon$, where ϵ is some random noise sampled from a gaussian distribution with mean zero and (unknown) standard deviation σ. We write $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

2.2.1 [10 points]. Write the log-likelihood of the observations $\{(x_1, y_1), \ldots, (x_N, y_N)\}$ as a function of w. *Hint*: recall that for the joint distribution $p(a, b)$ of two random variables a and b, we have $p(a, b) = p(a|b)p(b)$.

2.2.2 [10 points]. Show that maximizing the log-likelihood with respect to w is equivalent to minimizing the sum of the squared errors as in Equation (2).

3 Logistic regression [10 points]

In the gambling world, the "chance of winning" a game with two outcomes (gain or loss) are often stated as *odds*, defined as a ratio. For instance, a 7:1 odd means that the probability of winning is 7 times higher than the probability of losing. As often in machine learning, for values with many different possible orders of magnitudes, it is easier to consider the logarithm of this value. Assume we want to predict the log of the odd ratio with a linear model, such that

$$\mathbf{w}^\top \mathbf{x} \simeq \log\left(\frac{P(y=1)}{P(y=0)}\right) \tag{4}$$

Prove that predicting the log of the odd ratio with a linear model is equivalent to predicting the probability of winning by applying a sigmoid function to the output of a linear model, and that training our parameters by maximizing the log-likelihood of a set of observation gives us the logistic regression model. You may want to use the natural logarithm for this exercise.

4 Clustering [30 points]

4.1 K-Means [12 points]. In the K-Means algorithm, given N unlabeled points $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and a number of clusters K, we apply a greedy algorithm to find K cluster centers $\{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$ and a function $a : \mathbb{R}^D \to \{1, \ldots, K\}$ assigning a point $\mathbf{x}$ to one of the K clusters in order to minimize the inertia

$$\frac{1}{N} \sum_{n=1}^{N} \|\mathbf{x}_n - \mathbf{c}_{a(\mathbf{x}_n)}\|_2^2 \tag{5}$$

For the purpose of this exercise, we can rewrite this objective as

$$\frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} a_{n,k} \|\mathbf{x}_n - \mathbf{c}_k\|_2^2 \tag{6}$$

where $a_{n,k} = 1$ if point $\mathbf{x}_n$ is assigned to cluster k and 0 otherwise.
Prove that if the cost from Equation (6) is minimized, i.e. we are at the global minimum of the cost, then

$$\mathbf{c}_k = \frac{\sum_{n=1}^{N} a_{n,k} \mathbf{x}_n}{\sum_{n=1}^{N} a_{n,k}} \tag{7}$$

That is, the cluster centers $\{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$ are the means of the points assigned to the respective clusters. Note that this result should *not* depend on the actual heuristic (algorithm) we use to minimize to cost function. *Hint*: you may want to compute the partial derivative of something with respect to something else.

4.2 Hierarchical clustering and Levenshtein distance. The Levenshtein distance is a string metric measuring the difference between two sequences of characters as the number of character insertions, deletions or substitutions necessary to transform one sequence into the other.
4.2.1 [8 points]. Show that the Levenshtein distance is indeed a distance, using the mathematical definition of the term.

In hierarchical clustering, the Levenshtein distance can be used as an element-wise distance, i.e. as a distance between two points in the feature space. We also need to define a group-wise distance, i.e. a distance between two groups of elements. Given an element-wise distance $d(\mathbf{x}_m, \mathbf{x}_n)$ between two elements $\mathbf{x}_m$ and $\mathbf{x}_n$, the single-linkage criterion $D(\mathcal{C}_i, \mathcal{C}_j)$ between two clusters $\mathcal{C}_i$ and $\mathcal{C}_j$ is defined as

$$D(\mathcal{C}_i, \mathcal{C}_j) = \min_{\substack{\mathbf{x}_m \in \mathcal{C}_i \\ \mathbf{x}_n \in \mathcal{C}_j}} d(\mathbf{x}_m, \mathbf{x}_n) \tag{8}$$

4.2.2 [10 points]. Is the single-linkage criterion a distance in the general case? Prove the result if it is, and provide a counter-example if it is not.