Exercise 1 – ML Basics

Introduction to Machine Learning

Exercise 1: Car price prediction

Learning goals

1) Translate real-world problem into ML concepts
2) Use proper mathematical notation for those concepts

Imagine you work at a second-hand car dealer and are tasked with finding for-sale vehicles your company can acquire at a reasonable price. You decide to address this challenge in a data-driven manner and develop a model that predicts adequate market prices (in EUR) from vehicles' properties.

Only for lecture group B

Characterize the task at hand: supervised or unsupervised? Regression or classification? Learning to explain or learning to predict? Justify your answers.

Solution

We face a **supervised regression** task: we definitely need labeled training data to infer a relationship between cars' attributes and their prices, and price in EUR is a continuous target (or quasi-continuous, to be exact – as with all other quantities, we can only measure it with finite precision, but the scale is sufficiently fine-grained to assume continuity). **Prediction** is definitely the goal here, however, it might also be interesting to examine the explanatory contribution of each feature.

How would you set up your data? Name potential features along with their respective data type and state the target variable.

Solution

Target variable and potential features:

Variable	**Role**	**Data type**
Price in EUR	Target	Numeric
Age in days	Feature	Numeric
Mileage in km	Feature	Numeric
Brand	Feature	Categorical
Accident-free y/n	Feature	Binary
...	...	...

Assume now that you have data on vehicles' age (days), mileage (km), and price (EUR). Explicitly define the feature space $\mathcal{X}$ and target space $\mathcal{Y}$.

Solution

Let x_1 and x_2 measure age and mileage, respectively. Both features and target are numeric and (quasi-) continuous. It is also reasonable to assume non-negativity for the features, such that we obtain $\mathcal{X} = (\mathbb{R}_0^+)^2$, with $\mathbf{x}^{(i)} = (x_1^{(i)}, x_2^{(i)})^\top \in \mathcal{X}$ for $i = 1, 2, \dots, n$ observations. As the standard LM does not impose any restrictions on the target, we have $\mathcal{Y} = \mathbb{R}$, though we would probably discard negative predictions in practice.

You choose to use a linear model (LM) for this task. The LM models the target as a linear function of the features with Gaussian error term.

State the hypothesis space for the corresponding model class. For this, assume the parameter vector θ to include the intercept coefficient.

Solution

We can write the hypothesis space as:

$$\mathcal{H} = \{f(\mathbf{x} \mid \theta) = \theta^\top \mathbf{x} \mid \theta \in \mathbb{R}^3\} = \{f(\mathbf{x} \mid \theta) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 \mid (\theta_0, \theta_1, \theta_2) \in \mathbb{R}^3\}.$$

Note the **slight abuse of notation** here: in the lecture, we first define θ to only consist of the feature coefficients, with $\mathbf{x}$ likewise being the plain feature vector. For the sake of simplicity, however, it is more convenient to append the intercept coefficient to the vector of feature coefficients. This does not change our model formulation, but we have to keep in mind that it implicitly entails adding an element 1 at the first position of each vector.

Which parameters need to be learned? Define the corresponding parameter space Θ.

Solution

The parameter space is included in the definition of the hypothesis space and in this case given by $\Theta = \mathbb{R}^3$.

State the loss function for the i-th observation using $L2$ loss.

Solution

Loss function for the i-th observation: $L\left(y^{(i)}, f\left(\mathbf{x}^{(i)} \mid \theta\right)\right) = \left(y^{(i)} - \theta^\top \mathbf{x}^{(i)}\right)^2$.

Now you need to optimize this risk to find the best parameters, and hence the best model, via empirical risk minimization. State the optimization problem formally and list the necessary steps to solve it.

Solution

In order to find the optimal $\hat{\theta}$, we need to solve the following minimization problem:

$$\hat{\theta} = \arg\min_{\theta \in \Theta} \mathcal{R}_{\text{emp}}(\theta) = \arg\min_{\theta \in \Theta} \left(\sum_{i=1}^{n} \left(y^{(i)} - \theta^\top \mathbf{x}^{(i)}\right)^2 \right)$$

This is achieved in the usual manner of setting the derivative w.r.t. θ to 0 and solving for θ, yielding the familiar least-squares estimator.

Congratulations, you just designed your first machine learning project!

Exercise 2: Vector calculus

The whole exercise is only for lecture group A!

Learning goals

1. Understand how vector-valued functions work
2. Perform calculus on vectors and matrices

Consider the following function performing matrix-vector multiplication: $f(\mathbf{x}) = \mathbf{A}\mathbf{x}$, where $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{x} \in \mathbb{R}^{n \times 1}$.

What is the dimension of $f(\mathbf{x})$? Explicitly state the calculation for the i-th component of $f(\mathbf{x})$.

Solution

In computing $\mathbf{A}\mathbf{x}$ we multiply each of the m rows in $\mathbf{A}$ with the sole length-n column in $\mathbf{x}$, leaving us with a column vector $f(\mathbf{x}) \in \mathbb{R}^{m \times 1}$. Thus, we have $f : \mathbb{R}^{n(\times 1)} \to \mathbb{R}^{m(\times 1)}$.

The i-th function component $f_i(\mathbf{x})$ corresponds to multiplying the i-th row of $\mathbf{A}$ with $\mathbf{x}$, amounting to

$$f_i(\mathbf{x}) = \sum_{j=1}^{n} a_{ij} x_j,$$

with a_{ij} the element in the i-row, j-th column of $\mathbf{A}$.

Now, consider the gradient (derivative generalized to multivariate functions) $\frac{\mathrm{d}f(\mathbf{x})}{\mathrm{d}\mathbf{x}}$ (a.k.a. $\nabla_{\mathbf{x}} f(\mathbf{x})$).

What is the dimension of $\frac{\mathrm{d}f(\mathbf{x})}{\mathrm{d}\mathbf{x}}$?

Solution

The gradient is the row vector[1] of partial derivatives, i.e., the derivatives of f w.r.t. each dimension of $\mathbf{x}$:

$$\frac{\mathrm{d}f(\mathbf{x})}{\mathrm{d}\mathbf{x}} = \begin{pmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} & \dots & \frac{\partial f(\mathbf{x})}{\partial x_n} \end{pmatrix}.$$

[1] Pertaining to one of two conventions; we use the *numerator layout* here (transposed version = *denominator layout*).

Now, since f is a vector-valued function, each component is itself a vector of length m. Therefore, we have $\frac{\mathrm{d}f(\mathbf{x})}{\mathrm{d}\mathbf{x}} \in \mathbb{R}^{m\times n}$, given by the collection of all partial derivatives of each function component:

$$\frac{\mathrm{d}f(\mathbf{x})}{\mathrm{d}\mathbf{x}} = \begin{pmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \dots & \frac{\partial f_1(\mathbf{x})}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_m(\mathbf{x})}{\partial x_1} & \dots & \frac{\partial f_m(\mathbf{x})}{\partial x_n} \end{pmatrix}$$

This matrix is also called the *Jacobian* of f.

Compute $\frac{\mathrm{d}f(\mathbf{x})}{\mathrm{d}\mathbf{x}}$.

Solution

We have

$$\frac{\partial f_i(\mathbf{x})}{\partial x_j} = \frac{\partial\left(\sum_{j=1}^{n} a_{ij}x_j\right)}{\partial x_j} = a_{ij}.$$

Doing this for every element yields

$$\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix},$$

and we have $\frac{\mathrm{d}f(\mathbf{x})}{\mathrm{d}\mathbf{x}} = \frac{\mathrm{d}\mathbf{A}\mathbf{x}}{\mathrm{d}\mathbf{x}} = \mathbf{A}$.

Had trouble with this exercise?

- For the upcoming contents, you need to be able to handle **matrix-valued computations** (multiplication, transposition etc.) and also matrix-valued **calculus**.
- For more explanations and exercises, including a useful collection of rules for calculus, we recommend the book "Mathematics for Machine Learning" (https://mml-book.github.io/book/mml-book.pdf).