Homework 1: The Very Basics

Reno Malanga, Christian Arndt
Behavioral Data Science

January 30, 2025

Problem 1

Mark all statements which are FALSE.

- A research finding is always reproducible if it is published in a reputable, peer-reviewed journal.
- Many-analyst studies aim to assess the robustness of scientific conclusions by having multiple independent analysts or researchers work on the same data set.
- A random variable is continuous if its support is countable and there exists an associated probability mass function (pdf).
- Probability density functions have a lower bound of 0 and an upper bound of ∞.
- Handcrafted feature engineering is an example of representation learning.
- A multi-layer perceptron (MLP) with no hidden layers is equivalent to a linear model.
- The activation functions used in the hidden layers of an MLP must always be non-linear for the network to model non-linear relationships.
- Computational graphs include variables as nodes and operations as edges in a directed acyclic graph (DAG).

Problem 2

Replication Crisis

Problem Setup

How can theory building and statistical methodology synergize to support strong and reproducible psychological science?

Problem Solution

Theory building is about generating ideas and intuitions about models. Statistical methodology, on the other hand, concerns itself with creating models which can capture the general shape and complexity of reality (or, at least, some subset of reality in the form of a dataset).

In order to have strong and reproducible psychological science, both theory building and statistical methodology should be considered. For example, if we have a strong theory with a weak statistical methodology, the theory itself would be rendered obsolete; there's no sense in explaining models if those very models are incapable of capturing real relationships. Similarly, a strong statistical methodology might create wonderful models, but paired with a weak theory that cannot make meaningful statements about those models, we're left with no real intellectual takeaways about reality.

Problem 3

Maximum Likelihood Estimation

Problem Setup

You are given a dataset $\{x_1, x_2, \ldots, x_N\}$ consisting of n independent and identically distributed (IID) data points. Assume that data points are drawn from a Gaussian distribution with probability density function:

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

where:

- μ is the mean (unknown, to be estimated)
- σ^2 is the variance (fixed and known)

Write down the parametric model (i.e., the likelihood) for the dataset. Formulate the maximum likelihood estimator (MLE) problem and find the value of μ that maximizes the likelihood function. This is your maximum likelihood estimate μ_{MLE}.

Hint: differentiate the log-likelihood function $\ell(\mu)$ with respect to μ and set the derivative to 0. Solve for μ.

Problem Solution

With the assumption that our model follows a Gaussian distribution and the parameters μ and σ^2, our parametric model is as follows:

$$\begin{aligned} \theta &= (\mu) \\ p(x \mid \theta) &= p(x \mid \mu) \sim \mathcal{N}(\mu, \sigma^2) \\ p_{\mathcal{N}(\mu,\sigma^2)}(x) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \end{aligned}$$

Given the above pdf, the maximum likelihood estimator is:

$$\theta_{MLE} = \underset{\theta}{\text{argmax}} \left(\log \left(\prod_{n=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_n-\mu)^2}{2\sigma^2}\right) \right) \right)$$

We can take the log of the above likelihood estimator to remove the product and bring the exponent to the front as a coefficient.

$$\begin{aligned} &= \underset{\theta}{\text{argmax}} \left(\log \sum_{n=1}^{N} \left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_n-\mu)^2}{2\sigma^2}\right) \right) \right) \\ &= \underset{\theta}{\text{argmax}} \left(\sum_{n=1}^{N} -\frac{(x_n-\mu)^2}{2\sigma^2} \log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) \right) \\ &= \underset{\theta}{\text{argmax}} \left(\frac{\log\left(\sqrt{2\pi\sigma^2}^{-1}\right)}{2\sigma^2} \sum_{n=1}^{N} (x_n-\mu)^2 \right) \end{aligned}$$

We can find θ_{MLE} by differentiating the log-likelihood function with respect to μ, setting the derivative to 0, and then solving for μ.

Let the above log-likelihood function be ℓ. Let us first consider a single term, t, of the above sum:

$$\begin{aligned} \frac{d}{dx} t &= (x_i - \mu)^2 \\ &= x_i^2 - 2x\mu + \mu^2 \\ &= -2x_i + 2\mu \\ &= -2(x_i - \mu) \end{aligned}$$

In setting the derivative to 0, we can ignore the coefficient in front of the sum. We can see that by setting the derivative to be equal to 0, therefore, x_i and μ must be equal to each other.

Continuing to ignore the coefficient, we can see that the derivative of ℓ as a whole will be equal to 0 when the following equation is satisfied:

$$\begin{aligned}
\sum_{n=1}^{N}(x_n - \mu) &= 0 \\
x_1 - \mu + x_2 - \mu + \cdots + x_N - \mu &= 0 \\
\left(\sum_{n=1}^{N} x_n\right) - N\mu &= 0 \\
\mu &= \frac{\sum x_n}{N}
\end{aligned}$$

We can see that $\mu_{MLE} = \bar{x}$. This makes sense; intuitively, it a normally distributed parametric model will align best with the data when the model's mean is equal to the sample mean.

PROBLEM 4

GRAPHICAL MODELS AND COMPUTATIONAL GRAPHS

Problem Setup

In this problem, you will construct graphical models for probability distributions and computational graphs for functions. First, consider the following two distributions involving four variables A, B, C, and D.

1. $P(A, B, C, D) = P(A)P(B \mid A)P(C \mid B)P(D \mid C)$
2. $P(A, B, C, D) = P(A)P(B)P(C \mid A, B)P(D \mid C)$

For each of the distributions above, draw the corresponding graphical model and label the nodes (variables) and directed edges appropriately. Make sure to clearly indicate the conditional dependencies indicated by the structure of the graph. Next, consider the following two functions:

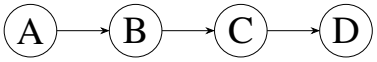
1. $f(x, y, z) = \log\left(\exp(x) + \sin(y) * z\right)$
2. $g(a, b) = \frac{1}{1+e^{-(a+b)}}$

For each function, write down the computational graph, showing the intermediate variables as nodes.

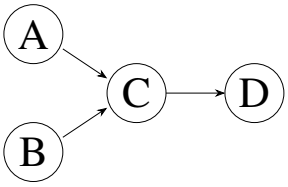
Problem Solution

Graphical Models

$P(A, B, C, D) = P(A)P(A \mid B)P(C \mid B)P(D \mid C)$

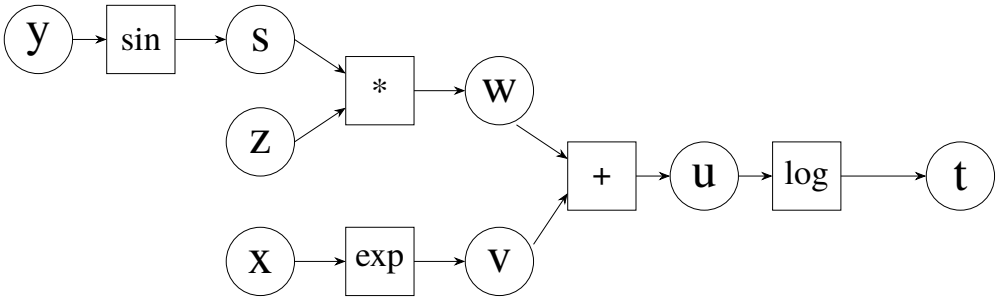


$P(A, B, C, D) = P(A)P(B)P(C \mid A, B)P(D \mid C)$



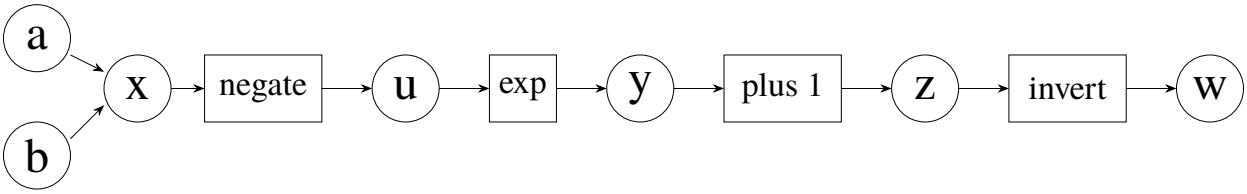
Computational Graphs

$f(x, y, z) = \log\left(\exp(x) + \sin(y) * z\right)$



$g(a, b) = \frac{1}{1+e^{-(a+b)}}$

Let **invert**$(x) = x^{-1}$.
Let **negate**$(x) = -x$.
Let **plus 1**$(x) = x + 1$.



PROBLEM 5

GIT AND GITHUB

Problem Setup

1. Create a public GitHub repository, create and add a team logo to the README file, along with some basic introductory notes on why behavioral data science is important for psychology and cognitive science. Create an `environment.yml` file and add all dependencies we have discussed so far. Then, in addition to the `main` branch, create separate branches for each of the two team members, from which you will be merging working code into the `main` branch.

2. Create a *merge conflict* (either for some of the coding exercises or a mock conflict) and resolve it.

3. Explain the differences between the following git commands:

   (a) `git restore`

   (b) `git checkout`

   (c) `git reset`

   (d) `git revert`

   in terms of undoing the changes to a repository by providing a minimal (actual or synthetic) example.

Problem Solution

Here is a link to our git repo for the semester's homework assignments: `https://github.com/christian-arndt/repo_men`

Our notes on why behavioral data science is important to psychology and cognitive science can be found in the repository's README file.

We each created a merge conflict in our own branches and resolved them independently.

1. `git restore` allows a user to set the state of the current working directory to the most recent commit. With the `--staged` flag, `git restore` can also be used to unstage files.

2. `git checkout` allows a user to switch between branches within repository or commits within a branch. `git checkout` does *not* reset the status of the branch; it simply moves the working head to the desired commit. We also used the `git checkout` command with the `-b` flag to create our own branches.

3. `git reset` allows a user to move the head of a branch to a desired commit *along with* the status of the branch. In contrast, `git checkout` only moves the head.

4. `git revert`: allows a user to set the branch status to a previous commit without removing any commits from the branches' commit history. It does this by creating a copy of the targeted commit as the most recent commit on the branch. In contrast, `git reset` removes the commit history when updating the branch status by moving the head and branch status backwards.

Problem 6

Python and NumPy

Problem Setup

In this exercise, you will write a Python program that approximates the value of π using Monte Carlo approximation, which we will cover in more detail next week. Your program should generate a sequence of random points and use these points to estimate the value of π. The accuracy of the program should improve as the number of points increases. Here are some hints:

- Your program should generate random points with x and y coordinates ranging from -1 to 1. This will simulate points within a 2×2 square that circumscribes the unit circle centered at the origin $(0, 0)$.
- For each generated point, determine whether it falls inside the unit circle. A point $p = (x, y)$ is inside the circle if $x^2 + y^2 \leq 1$.
- Use the ratio of the number of points that fall inside the circle (n_{inside}) to the total number of generated points (n_{total} to approximate π. The formula is given by:

$$\pi \approx 4 \times (n_{inside}/n_{total}) \tag{2}$$

Problem Solution

See the code in our github repository (`https://github.com/christian-arndt/repo_men`) under `homeworks/homwork_1`.

Problem 7

Linear Estimation for XOR

Problem Setup

In class, we discussed the impossibility of solving the XOR problem with a linear model of the form:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \tag{3}$$

where $(x_1, x_2) \in \{(0, 0), (0, 1), (1, 0), (1, 1)\}$. Show that extending the model with the interaction term $x_1 x_2$, that is,

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3, \tag{4}$$

leads to a solution and derive the resulting values for the β weights. This is a typical example of *feature engineering* by extending a linear model with a non-linear combination of the predictors.

Problem Solution

First, let us establish the $\oplus$ table for two variables:

x_1	x_2	$\oplus$
0	0	0
0	1	1
1	0	1
1	1	0

Now, let us define our linear model given inputs $\{x_1, x_x\}$ and parameters $\{\beta_0, \beta_1, \beta_2, \beta_3\}$:

$$x_3 = x_1 * x_2$$
$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$$

A solution is:

$$\beta_0 = 0$$
$$\beta_1 = 1$$
$$\beta_2 = 1$$
$$\beta_3 = -2$$

We can check this solution by plugging the β-values into our model and checking that the outputs are identical to those defined in the $\oplus$ table. Our model with the β-values plugged in is:

$$y = x_1 + x_2 - 2x_3$$

Using this equation, our model outputs the following table:

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

This table is identical to the $\oplus$ table, whcih means our model now solves XOR.