Statistical Theory and Modeling, 7.5 hp

Home assignment - Part 2

Mattias Villani

2025-04-30

Table of contents

Problem 4 - Linear regression in vector form 1
Problem 5 - Numerical maximum likelihood for negative binomial distribution 3
Problem 6 - Regression models for count data 4
Bonus problem 6

> Note
>
> This is the second part of the home assignment for the course. The first part had Problems 1-3, so I will continue with this numbering, and the first problem here will therefore be Problem 4.

Problem 4 - Linear regression in vector form

Problem 4a)

Let $a = (1, 0, 2)^\top$ and $b = (1, 0, -1)^\top$ be two vectors. The dot product is computes in R as

```r
a = c(1,0,2)
b = c(1,0,-1)
a%*%b
```

```
     [,1]
[1,]   -1
```

Note that R returns the dot product as a matrix (with one row and one column) even though it is a scalar (a number). If you really want a number you can do return the first (and only) element like this:

```r
(a%*%b)[1]
```

```
[1] -1
```

Are a and b orthogonal?

Problem 4b)

Simulate a 10×3 matrix X with standard normal N(0,1) random variables. Let $\beta = (1, 1, 2)^\top$ be a vector. Compute the matrix-vector product $\mu = X\beta$. When you code this, use the variable name `mu` for μ and `b` for β. Explain how the *first* element of μ relates to the elements in X.

Problem 4c)

Now simulate a vector of errors ε (use the variable name `epsilon`) from a normal distribution with mean zero and standard deviation $\sigma = 0.1$. Compute the vector of response observations $y = X\beta + \varepsilon$. Compute the least squares estimate $\hat{\beta} = (X^\top X)^{-1} X^\top y$ based on the simulated X and y.

Problem 4d)

The variance of the errors in ε can be estimated from the vector of *residuals* $e = y - X\beta$ as follows

$$s^2 = \frac{e^\top e}{n - p}$$

This is the same residual variance formula as you used in the previous statistics course, where it was written using sums instead of vectors

$$s^2 = \frac{\sum_{i=1}^n e_i^2}{n - p}$$

The covariance matrix of the least squares $\hat{\beta}$ estimator can be estimated by the matrix formula

$$s^2(X^\top X)^{-1}$$

and the standard errors for each $\hat{\beta}$ is therefore the square root of the diagonal elements of this covariance matrix. Compute the standard errors of the regression coefficients based on the 10 observations that you simulated in Problems 4c).

Problem 5 - Numerical maximum likelihood for negative binomial distribution

Problem 5a)

We continue with the bugs data from the first part of the Assignment. The file `bugs.csv` contains a dataset with the number of bugs and some other explanatory variables for $n = 91$ releases of several software projects. We load the data:

```
data = read.csv("https://github.com/StatisticsSU/STM/raw/main/assignment/bugs.csv",
                header = TRUE)
y = data$nBugs # response variable: the number of bugs, a vector with n = 91 observations
X = data[,-1]  # 91 x 5 matrix with covariates
X = as.matrix(X) # X was initial a data frame, but we want it to be matrix
```

In this problem you will learn to find the maximum likelihood estimate numerically for models with a single parameter. We will only use the response variable y, the number of bugs. Later in this assignment we will also use the covariates/features in matrix $\mathbf{X}$.

We use the negative binomial again, but this time we try to estimate the parameter r given that the mean parameter μ is fixed at the estimate $\hat{\mu} = \bar{y} \approx 5.2528$. So the model is

$$Y_1, Y_2, \ldots, Y_n \overset{iid}{\sim} \text{NegBin}(r, \mu = 5.2528)$$

and your task is to find the maximum likelihood estimate $\hat{r}$ of the parameter $r > 0$ using numerical maximization in R. Set the argument lower to some small, but positive value, for example `lower = 0.0001`. `lower = 0` will not work since the negative binomial distribution with $r = 0$ is not defined.

Problem 5b)

What is the standard error of the ML estimate $\hat{r}$ in Problem 5a)? Recall that the standard error is the standard deviation in the sampling distribution of an estimator. An approximate answer is enough, and you can obtain it numerically (no maths needed!).

Problem 5c)

Let us now try to estimate both parameters μ and r at the same time (jointly, as we say) using the maximum likelihood method. Use numerical maximization again and set lower = c(0.0001, 0.0001) to make sure that the estimated parameters are positive, as they should be in the negative binomial distribution.

Problem 5d)

Compute the standard errors for $\hat{r}$ and for $\hat{\mu}$.

Problem 6 - Regression models for count data

Ok, now we make this more interesting by modeling the expected number of bugs *as function of covariates/features* in the `X` matrix loaded above:

- *intercept* - this is columns of ones to model the intercept
- *nCommit* - the number of commits since the last release of the software
- *propC* - proportion of C/C++ code in the project
- *propJava* - proportion of Java code in the project
- *complexity* - a measure of code complexity that takes into account the frequency of if statements etc.

Problem 6a)

Use numerical optimization to fit the **Poisson regression model**

$$Y_i | x_i \overset{\text{ind}}{\sim} \text{Poisson}\Big(\lambda_i = \exp\left(x_i^\top \beta\right)\Big)$$

Note how each observation has its "own" λ_i parameter, which is modeled as the exponential of $x_i^\top \beta$. We use the exponential function to make sure that λ_i is always positive, as it has to be in the Poisson model. Note that a Poisson regression is a model for the **conditional expectation**

$$\mathbb{E}(Y_i | x_i) = \lambda_i = \exp\left(x_i^\top \beta\right) = \exp\left(\beta_0 + \beta_1 x_{1,i} + \ldots + \beta_p x_{p,i}\right).$$

The covariates/features for the ith observation $x_i = (x_{1,i}, \ldots, x_{p,i})^\top$ is given the ith row of the matrix `X` . For example, for the second observation we have `y[2] = 6`, so six bugs in the second release, and the covariate values for this second release are:

```r
X[2,]
```

```
 intercept   nCommits      propC   propJava complexity
 1.0000000  4.0000000  0.4150649  0.3295860  0.7031914
```

That is, this release (observation) has 4 commits, approximately 41.4% C code, 32.9 % Java code and a Code Complexity of 0.7. The regression coefficients in $\beta = (\beta_0, \beta_1, \ldots, \beta_4)^\top$ measures how the mean of the Poisson distribution λ_i changes when the features change. For example, β_4 determines how sensitive the mean number of a bugs in a release is to the code complexity. However, since λ_i is not a linear function of the features (as a result of the exponential function), the β coefficients cannot be interpreted as in ordinary linear regression; see the Lecture on Poisson regression.

Ok, I can sense that you are eager to fit this model, so let's do it! Use the `bugs` dataset to compute the maximum likelihood estimate of $\beta = (\beta_0, \beta_1, \ldots, \beta_4)^\top$ with numerical optimization. Note that since we have used the exponential function in the model, the elements of $\beta = (\beta_0, \beta_1, \ldots, \beta_4)^\top$ are free to take on any value, also negative values. So you can use `method = c("BFGS")` and there is no need to use the `lower` argument in optim.

Problem 6b)

Use the optimization output from the previous problem to get approximate standard errors for each estimated β coefficient. Compute 95% confidence intervals for each β parameter. Which covariates seem to be statistically different from zero? [Hint: use the large-sample normal approximation of the sampling distribution, and use the confidence interval to check the statistical significance.]

Problem 6c)

The upcoming release, which we here number as observation number 92, has the following covariate vector:

```r
x92 = c(1, 10, 0.45, 0.5, 0.89)
```

So, the release is based on 10 commits, good proportions of C and Java code and a high code complexity of 0.89. What is the predicted number of bugs for this release?

Bonus problem

> Caution
>
> This is a bonus problem. It is not very hard, but do it only if you feel up for it and have energy left. You also need time to study for the exam. And to have a life.

Consider the negative binomial regression

$$Y_i|x_i \overset{\text{ind}}{\sim} \text{NegBin}\Big(r, \mu_i = \exp\left(x_i^\top \beta\right)\Big)$$

The parameters in this model is therefore r and the regression coefficients in β. Find the maximum likelihood estimates of the parameters using numerical optimization for the `bugs` data. Note that you want to use `lower = (0.0001, -Inf, -Inf, -Inf, -Inf, -Inf)` here to make sure that r is positive and the regression coefficients are unrestricted.