Statistical Data Analysis II

Project 1, deadline: 08.12.2022 10:15 am

Project overview

The goal of Project 1 is to apply different Variational Autoencoders (*VAEs*) to biological data. We will do this by customizing a VAE using probabilistic modeling.

Data

Single cell RNA sequencing. In this Project, we will be using data from single cell RNA sequencing (*scRNA-seq*). Organs such as the pancreas are made up of many types of tissues and these tissues are made up of many types of cells. For example, within the pancreas, we can identify cells that are typical of this organ, such as alpha or beta cells, but also cells related to the blood supply or the immune system. It can be difficult to study all of these different cell types with traditional methods, but scRNA-seq provides the high resolution needed to identify and study each individual cell type. This is important for understanding specific functions of the tissues (e.g. pancreas) and how it exactly works. Among other tasks, it is possible to compare pathological cells, taken from cancer patients, with healthy cells.

The result of a *scRNA-seq* experiment is a matrix of counts in which the rows correspond to cells and the columns correspond to genes. We will refer to this as gene expression matrix. Each entry in this matrix is a count of mRNA molecules present in a given cell (alert: major biological oversimplification here). Loosely speaking, this can be interpreted as gene activity. In plain English, the higher the count of a given gene in a given cell, the more active it is within the cell.

Dataset. The dataset was collected from the bone marrow of human donors. The cells collected are mostly cells of the immune system. This dataset was released as a part of NeurIPS 2021 "Open problems in Single Cell Analysis" competition.

How to download the dataset? The data can be downloaded from GDrive. You have access to two files containing gene expression matrices for training and test, respectively.

How to work on the project

Your Project should live in a GitHub repository. Please send a link to your repository by e-mail to your TA: *p.szymczak8@uw.edu.pl* or *adam.izdebski@uw.edu.pl*, if you haven't already done that.

What should your repository contain? Your repository should contain the following files:

- **A report** - this is the most important file. We will determine the number of points solely based on this file. Hence, it should be visible from the report, without looking up the source code, how did you handle the most crucial parts of the given exercise. The tasks are structured by words: explain, plot, or show. For example, if your task is to modify the decoder, you ought to show this 45555555555modification and only the modification (it will be easier to describe it via mathematical formulation or code). If your task is to show a plot, then show the plot, but do not show the code that generated the plot. Please name your report *SAD2022Z_Report_[NameSurname].pdf*. Tip: It is very easy to convert *.ipynb* into *.pdf*. All the figures, plots and results should be imported from your drive or repository. No computations should be required to compile the report.
- **Scripts** We want to put an emphasis on project reproducibility. In the GitHub repo, apart from the .pdf file with the report, we expect at least three files:
  - *eval.py* - the point of this script is to reproduce all the figures, plots and results so that they can appear in the report pdf. This should take pre-trained models and produce results, figures and plots.
  - *train.py* - this script trains and saves all the models. You can include one script for one class of a model.
  - *environment.yaml / .txt* - a conda requirements file.

Tip: you can (should) use argparser and bash as in the research handbook from our GitHub resources.

Grading

You can receive 99 points for the entire project. The maximum number of points for each of the subsections is given in parentheses. Number of points from Project 1 will be based on **the content and the style of the report**. Write specific and concise

text. Remember that each figure should be captioned and titled, and all axes should be labeled. Report should be formatted so that it is easy to read.

The project should be reproducible. It should be possible to clone your repo, setup a conda environment using the *environment.yaml* file, train the models using *train.py*, and then produce the figures etc. using *eval.py* script. This is a necessary condition.

Collaborating. We are aware that solutions will be discussed in small groups and pieces of code shared. While we encourage you to discuss and try to help each other out, we strongly discourage sharing final reports or pieces of code. We want to avoid a situation where the same argumentation will appear in different reports and the same small typos will appear in different repositories. But do discuss solutions and try to help each other out!

Deadline. Please do not commit to your repository after the deadline.

Why is task 4. b worth only 10 pt? We know that altogether this is a lot of things to do, so you can omit this point and still get a good grade ($< 90\%$).

Tasks

A general tip is to play around with a small subset of data (e.g. 1000 cells) first and only after you finish prototyping run models and other experiments on all data.

1. Exploration *(19 pt.)*
In this Task it will be helpful to use `anndata`. It is a package used to operate on `.h5ad` objects that is commonly used in scRNA-seq analysis. Quick tutorial here. We will be dealing with sparse matrices, but you can always cast it to dense (watch out for OOM) using the method $\texttt{sparse}_m atrix.toarray()$.

(a) *(1 pt.)* Load your dataset using `adata = sc.read_h5ad(path)` and report how many observations and variables the loaded training and test data sets contain.

(b) *(3 pt.)* Access the `adata.X` object, which contains a matrix of counts which has been already preprocessed, and the `adata.layers['counts']` object, which contains raw data. Plot histograms of both the raw data and the processed data. Pay attention to the X-axis and the range of values spanned by the data.

(c) *(4 pt.)* Explain what kind of preprocessing has been applied to the preprocessed matrix. *Tip: You can also check min, max, mean, etc.*

(d) *(3 pt.)* Remove zeroes from both the raw and the processed matrices. Plot histograms of both modified data sets.

(e) *(5 pt.)* Explain what is the distribution of the data. Explain what the abundance of zeroes means in this context? *Tip: There is an biological explanation.*

(f) *(3 pt.)* Access the `adata.obs` object. Explain what is the information contained in this data frame. Report the the number of patients, the number of labs, and the number of cell types in the data.

2. Vanilla VAE training *(30 pt.)*

In this task you will adjust the VAE model you implemented during Lab 6 and Lab 7 so that it can be trained on scRNA-seq data. The input to the model (and the output) should be the gene expression matrix. You need to decide whether you want to use the raw or the preproccesed matrix (`adata.X` or `adata.layers['counts'].X`). Note that your VAE should have a stochastic Encoder, Decoder and be trained with a probabilistic loss.

(a) *(10 pt.)* Train a VAE model on the data. Verify the training procedure by showing learning curves for both training and test sets. *Reminder: By a learning curve we mean plotting the -ELBO against epoch numbers.* Additionally, break down ELBO into reconstruction and regularization losses and make a plot accordingly.

(b) *(10 pt.)* Fit a PCA to the latent space. What number of principal components explains more than 95% of the variance? Show a table in which you choose three different latent space sizes and report -ELBO on the test set for each of the models.

(c) *(5 pt.)* Use a PCA to map all the encoded cells from the test set onto the top two principal axes and show plots for all models from the previous point. *Tip: If the PCA plot collapses, try subsampling the observations to a smaller number.* Extend the PCA plots by colouring each point with `adata.obs.cell_type`.

(d) *(5 pt.)* Choose the final model, explain your decision making process behind choosing the data set (raw or preprocessed) and choosing the size of the latent space.

3. Custom Decoder *(35 pt.)*

In this task you will modify the Decoder of your VAE to better fit the distribution of the data.

(a) *(10 pt.)* Using insights from Task 1 Exploration, implement a Decoder that models the data distributed according to a sensible distribution. Explain you reasoning behind choosing the Decoder.

(b) *(20 pt.)* Train a VAE with the new Decoder on data selected in the previous task. Plot learning curves for the new model alongside the final model chosen in the previous task. Note that here again you need to make an educated guess on the size of the latent space. Is your new Decoder better?

(c) *(5 pt.)* Plot the latent space of your final model from this task and from the previous task using PCA, as done in the previous exercise. Compare the plots. Explain whether you see any differences.

4. Adjusting VAE for batch effect *(15 pt.)*

The batch effect is a common and serious problem in scRNA-seq. Shortly, cells from different sources, that are supposed to be similar to each other, are not close in the low dimensional representation space. This creates the false impression that there are many distinct groups of cells, but there are not. The main reason for this is that the gene expression values are measured in different conditions, e.g. different batches, times or different technicians.

In a dataset without the batch effect, we hope for a low dimensionality embedding where the visual grouping (or even clustering) of the cells is guided solely with biological signal (e.g. cell type). In turn, in a dataset where the batch effect is present, the cells are visually grouped by lab/patient/other factors. The batch effect can lead to incorrect conclusions about the underlying biological mechanism, as the comparison between batches may be confounded by the batch effect.

In this task you will check whether we should be worried about batch effect in our dataset and how to address such case by extending the VAE framework.

(a) *(5 pt.)* Extend the PCA plots from both Vanilla VAE and VAE with a custom decoder so that the points are coloured by `adata.obs.batch`, `adata.obs.DonorID`, and, most importantly, `adata.obs.Site`. Include the figures in the report.

(b) *(10 pt.)* Extend your VAE with a custom Decoder so that **either**:

(I) The model accepts a vector of site assignments that goes through encoder

(II) The model concatenates a vector of site assignments to the vector of latent space, and the resulting vector goes through decoder.

Analyze the model and its latent space according to steps outlined in previous tasks.