ELBO Challenge
Single-Cell ANNotation using Variational Inference

Romain

November 2, 2018

Last year, I asked prospective interns to derive the *evidence lower bound* (ELBO) in the case of my very first model (named ZINB-VAE [1]) since a lot of implementation details were not given in the manuscript, especially the parametrization of the negative binomial distribution and some numerical stability tricks. All these details are now present in our main manuscript of the single-cell Variational Inference model (scVI) [2] on bioRxiv (see supplementary materials). In this new assignment, we explore into further details the extension of scVI for cell-type annotation (I am writing a journal paper on this work that will serve as a "solution" :p)

The Bayesian inference problem solved by scVI [2] falls into the category of unsupervised learning problems. This essentially means that we can observe some gene expression data x that we would like to summarize into a latent vector z. The extension of scVI we are interested in this assignment is to take into account some annotation of the cells (provided by a domain expert such as a computational biologist) c. In the case of a prediction problem for handwritten digits x (resp. c) would be the image encoded in pixel space (resp. the number in the image). However, labeling cells (or images) is a painful task and it makes sense to consider a scenario where one sometimes observes the pair (x, c) and sometimes only the gene expression x. We are therefore interested in building a generative model like scVI that can perform well at predicting cell-types c in the semi-supervised learning framework. For that, we relate to recent work [3] and adapt it for scVI.

In this assignment, we will provide students with the full generative model (which has been carefully designed in around six months of work). Your task will be to derive the ELBO as intelligibly as possible by using notations from previous papers such as [2, 3, 4] and link as much as possible your objective function and parameters to the code in [5]

General setting

We consider the problem of working with outputs of K different scRNA-seq experiments. An experiment k contains primarily an $N_k \times G$ matrix where each entry records the number of transcripts measured for each of the G genes in each of the N_k cells. Each experiment also summarizes the main biological properties of interest in some form of annotation, either discrete cell types or continuous differentiation paths encoded in a N_k vector. In this work, we format our data as a matrix $(x_{ng})_{n\in\{1,\cdots,\sum_k N_k\},g\in\{1,\cdots,G\}}$ with dataset-identifier covariate $\gamma_n \in \{1, \cdots, K\}$ and possibly annotation c_n. In this manuscript, c_n will describe discrete cell types only (we leave more complex structure over labels such as tree or gradients as a future research direction). For clarity, we also make no distinctions between datasets and batches in this work.

As in [2], we build a deep generative model where the expression level x_{ng} is zero-inflated-negative binomial (ZINB) when conditioned on γ_n the batch identifier , ℓ_n a hidden nuisance factor accounting for variations in capture efficiency and sequencing depth and $z_n^{\mathbf{1}}$, the remaining variability expected to reflect biological differences between cells. The essential innovation lies in the more refined hierarchical structure we bring to this random variable $z_n^{\mathbf{1}}$ which we make depend of the cell annotation c_n, which can be either observed or hidden, in a semi-supervised fashion. We name the resulting model SCANVI (Single Cell ANnotation using Variational Inference).

One typical scenario of interest deals with the problem of annotating a unlabeled experiment from a previously published experiment with annotated labels. This task is essentially a domain-adaptation task treated in the statistical literature. Another scenario is where all the datasets have labels but only partially. This happens for example when only high quality cells are kept in

a study. Furthermore, we note that SCANVI is a Bayesian model and provide an full posterior uncertainty in the labels and not only a point estimate. That can be particularly useful when labels cannot be entirely trusted as it is the case in our experiments (labels are computationaly derived).

An extension to scVI for semi-supervised cell-type annotation

SCANVI is a hierarchical Bayesian model for single-cell RNA sequencing data with conditional distributions parametrized by neural networks. SCANVI's probabilistic graphical model encodes conditional independence assumptions necessary to disentangle batch-effects and biological signal. In our generative model, we assume each cell n is an independent realization of the following generative process. Latent variable

$$z_n^{\mathbf{2}} \sim \text{Normal}(0, I)$$

is a low-dimensional random vector describing cell n. Let $\mathbf{c}$ describe the expected proportion of cells for each cell-types. Latent variable

$$c_n \sim \text{Multinomial}(\mathbf{c})$$

describes the cell-type of the cell n. By combining cell-type information c_n and random vector $z_n^{\mathbf{2}}$, we create a new low-dimensional vector

$$z_n^{\mathbf{1}} \sim \text{Normal}(f_{z^{\mathbf{1}}}^{\mu}(z_n^{\mathbf{2}}, c_n), f_{z^{\mathbf{1}}}^{\sigma}(z_n^{\mathbf{2}}, c_n))$$

where $f_{z^{\mathbf{1}}}^{\mu}$ and $f_{z^{\mathbf{1}}}^{\sigma}$ two functions parametrized by neural networks. Given ℓ_μ and ℓ_σ^2 specified per batch, latent variable

$$\log \ell_n \sim \text{Normal}(\ell_\mu, \ell_\sigma^2)$$

encodes a cell-specific scaling factor. Let θ encode a gene specific inverse dispersion parameter. Given the batch information γ_n, conditional distribution $x_{ng} \mid z_n^{\mathbf{1}}, \ell_n, \gamma_n$ is conform to the one from the scVI model

$$\begin{aligned}
w_{ng} &\sim \text{Gamma}(f_w(z_n^{\mathbf{1}}, \gamma_n), \theta_g) \\
y_{ng} &\sim \text{Poisson}(l_n w_{ng}) \\
h_{ng} &\sim \text{Bernoulli}(f_h(z_n^{\mathbf{1}}, \gamma_n)) \\
x_{ng} &= \begin{cases} y_{ng} & \text{if } h_{ng} = 0, \\ 0 & \text{otherwise.} \end{cases}
\end{aligned}$$

where f_w and f_h are functions parametrized by neural networks. In this model, annotation c_n can be either observed or unobserved following [3, 4], which is useful in our applications where some datasets would come partially labeled or unlabeled. Only the first part of the generative model, as separated above, differs from the original scVI formulation. This corresponds to the top part of the new representation of the graphical model Figure 1.

Efficient approximate posterior inference with neural networks

Because the distribution $p(x_{ng}, c_n \mid \gamma_n)$ if c_n observed (resp. $p(x_{ng} \mid \gamma_n)$ otherwise) is not amenable to exact Bayesian computation, we use variational inference parametrized by neural networks [6]. As in [2], we integrate the random variables $\{w_{ng}, y_{ng}, h_{ng}\}$ to simplify our model ($x_{ng} \mid z_n^{\mathbf{1}}, \ell_n, \gamma_n$ is Zero-Inflated Negative Binomial) and use variational inference and neural networks to perform efficient approximate inference [6] over the latent variable $\{z_n^{\mathbf{1}}, z_n^{\mathbf{2}}, c_n, \ell_n\}$. We assume our variational distribution has the following form:

$$q_\Phi(z_n^{\mathbf{2}}, c_n, z_n^{\mathbf{1}}, \ell_n \mid x_n, \gamma_n) = q_\Phi(z_n^{\mathbf{2}} \mid z_n^{\mathbf{1}}, c_n, \gamma_n) q_\Phi(c_n \mid z_n^{\mathbf{1}}) q_\Phi(z_n^{\mathbf{1}} \mid x_n, \gamma_n) q_\Phi(\ell_n \mid x_n, \gamma_n)$$

Following [3, 4], we derive two variational lower bounds: one $\mathcal{L}$ in the case of c_n observed for $p_\Theta(x_n, c_n \mid \gamma_n)$ and a second $\mathcal{U}$ in the case of c_n non-observed for $p_\Theta(x_n \mid \gamma_n)$ where Θ are all the parameters (neural networks and inverse-dispersion parameters). We optimize the sum

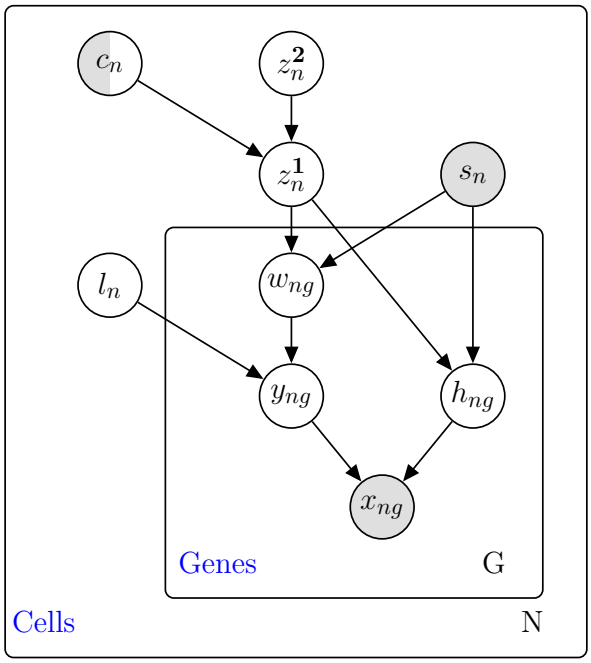

Figure 1: The underlying graphical model of SCANVI. Shaded vertices represent observed random variables. Semi-shaded vertices represent variables that can be either observed or random. Empty vertices represent latent random variables. Edges signify conditional dependency. Rectangles ("plates") represent independent replication.

ELBO $= \mathcal{L} + \mathcal{U}$ over the neural networks parameters. Remarkably, the approximate posterior $q_\Phi(c_n \mid z_n^{\mathbf{1}})$ can be used as a classifier, assigning cells to cell-types based on their location in latent space. So that this classifier also learns from labeled data, we add an additional cross-entropy term to the ELBO as in [3, 4].

We sample from the variational posterior using the reparametrization trick [6] as well as "mini-batches" from the dataset to compute unbiased estimate of the objective gradients' with respect to the parameters. We use Adam as a first-order stochastic optimizer to update the model's parameters.

Assignment for students

We wrote in this manuscript the details of the SCANVI model and its inference procedure. Your task is to write all the equations that leads to the celebrated *evidence lower bound* (or *variational lower bound*) and map these terms to the code our scVI team implemented last summer. Even though the exact equations have never been written, you can derive them yourself following the research material cited in this assignment.

1. Let us assume that c_n is observed. Derive the *evidence lower bound* (ELBO) for the marginal probability $\log p(x_n, c_n \mid \gamma_n)$ by applying Jensen inequality with the variational distribution $q(\ell_n, z_n^{\mathbf{1}}, z_n^{\mathbf{2}} \mid \gamma_n, x_n, c_n)$.

2. Let us assume now that c_n is not observed. Derive the ELBO for the marginal probability $\log p(x_n \mid \gamma_n)$ by applying Jensen inequality with the variational distribution $q(\ell_n, z_n^{\mathbf{1}}, z_n^{\mathbf{2}}, c_n \mid \gamma_n, x_n)$.

3. Simplify your notations as much as possible by using KL divergences and map your expression to the implementation in [5]. In particular, rigorous comparisons and honest answers (saying "I don't know why this line of code is here" is alright) will be highly appreciated.

Quick comments to guide your work

The difficulty of this assignment essentially comes from:

1. quickly understanding the principles of:
   - Bayesian networks [7]
   - Variational Inference [8]
   - its use case with neural networks approximation to the posterior [6]

   You can train your ELBO derivations skills by working with simpler cases. Especially, think about neural networks as simple functions.
2. the fact that you need to marginalize out some of the random variables before applying VI in our biology application [2] (This is fully explained in appendices, no modification is needed in your case).
3. the semi-supervised learning framework, explained in [3, 4]
4. the idea that we condition on the batch (dataset) identifier [2, 4]
5. Reading and understanding variational autoencoders implementation details. You can read simpler codes such as simples VAEs [9] in PyTorch as well as our scVI code on which SCANVI is built (`VAE` class)
6. Finally, there are tons of details in our repo that are specific to the biology application. In particular, we expect you to not figure out all of them. Especially, we **strongly advise** you to either not treat or focus later on:
   - `log_variational` is here for numerical stability
   - `labels_groups` is for hierarchical classification, do not look at any lines of code that include this (this is not explained in this PDF).
   - `dispersion` indicates how some parameters for the negative binomial are treated. You should ignore this (the scVI manuscript describes the case `dispersion = "gene"`)
   - How batch identifiers γ_n are treated requires good knowledge of PyTorch. You should not worry about `n_cat_list` as a first step.
   - The additional cross-entropy we refer to in this PDF is added the `SemiSupervisedTrainer` class. You should comment on this only if you explained well the inference procedure.

References

[1] "ZINB-VAE," 2017. [Online]. Available: https://arxiv.org/pdf/1709.02082v1.pdf

[2] R. Lopez, J. Regier, M. B. Cole, M. Jordan, and N. Yosef, "Bayesian Inference for a Generative Model of Transcriptome Profiles from Single-cell RNA Sequencing," *bioRxiv*, 2018. [Online]. Available: https://www.biorxiv.org/content/early/2018/03/30/292037

[3] D. P. Kingma, S. Mohamed, D. Jimenez Rezende, and M. Welling, "Semi-supervised Learning with Deep Generative Models," in *Advances in Neural Information Processing Systems 27*, 2014, pp. 3581–3589.

[4] C. Louizos, K. Swersky, Y. Li, M. Welling, and R. Zemel, "The Variational Fair Autoencoder," in *International Conference on Learning Representations*, 2016.

[5] "scanVI code," 2018. [Online]. Available: https://scvi.readthedocs.io/en/master/_modules/scvi/models/scanvi.html

[6] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *The International Conference on Learning Representations*, 2014.

[7] "An Introduction to Probabilistic Graphical Models," 2003. [Online]. Available: http://people.eecs.berkeley.edu/~jordan/prelims

[8] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[9] "An example of PyTorch implementation of VAEs." [Online]. Available: https://wiseodd.github.io/techblog/2017/01/24/vae-pytorch/