Summary and discussion of: “ZIFA in single-cell data analysis”

CASIA report

Li Donghao 20653877

1 Quick summary

In this report, I explored how to do dimensional reduction for zero-inflated data which means there are many zeros items. First, Zero Inflated Factor Analysis model [2] is introduced and EM algorithm is used to estimate the parameters. Simulation study and real-world experiments show its effectiveness. Then, VAE models [1] are also adopted for this problem. I implement two types of VAEs: original VAE with Gaussian or poisson decoder and “Zero-inflated VAE”. “Zero-inflated VAE” is motivated by ZIFA, I tried to modify the original VAE to handle zero-inflated data. Specifically, it has two decoder: a guassian decoder to approximate non-zeros items and a Bernoulli encoder to simulate the dropout events.

Experiments show the VAE with Poisson decoder outperforms all the others significantly. While ZIFA also gets a decent performance. However, experiments show this does not provide any advantage to the original VAEs. Finally, some discussion is included and there is still some potential work that can be done in adopting VAE for process zero-inflated data.

There is one more thing to emphasize. The ZIFA implementation is copied from https://github.com/epierson9/ZIFA. My own implementation failed to give correct solution due to some unknown reasons. However, I also uploaded on my github.(see zifa.py)

2 Derivation of ZIFA

2.1 Model assumptions

Let Y be a N by D single cell RNA expression data after log(count+1) transformation. Since there exists dropout events, there are many zero items in Y. To do dimensional reduction, “ZIFA Zero-Inflated Factor Analysis” is proposed by [2]. They assume that Y_i is generated from a low dimensional vector Z_i, the generation process is followed.

$$
\begin{aligned}
z_i &\sim N(0, I),\\
x_i|z_i &\sim N(Az_i + \mu, W),\\
h_{ij}|x_{ij} &\sim Ber(p_{i,j}), \quad where p_{i,j} = exp(-\lambda x_{ij}^2)\\
y_{ij} &= x_{ij} \quad when \quad h_{ij} = 0\\
y_{ij} &= 0 \quad when \quad h_{ij} = 1
\end{aligned}
$$

In this model, dropout event is captured by H, and the probability of dropout event is related with X. In the following derivation, we use y_{i+} to donate non-zero elements of y_i, y_{i0} to donate zero elements of y_i (same for x_i).

2.2 Derivation of EM algorithm

First we write down the complete data likelihood:

$$\begin{aligned} p(Z, X, H, Y|\Theta) &= \prod_{i=1}^{N} \left[p(z_i) \prod_{j=1}^{D} p(x_{ij}|z_i) p(h_{ij}|x_{ij}) p(y_{i,j}|x_{i,j}, h_{i,j}) \right] \\ &= \prod_{i=1}^{N} \left[p(z_i) \prod_{j:y_{ij}=0} p(x_{ij}|z_i) p(h_{ij}=1|x_{ij}) \prod_{j:y_{ij}\neq 0} p(x_{ij}=y_{ij}|z_i) p(h_{ij}=0|x_{ij}) \right] \end{aligned}$$

Next we write down the log likelihood:

$$\begin{aligned} l(A, \mu, W, \lambda) = \sum_{k=1}^{d} \log N\left(Z_{ik}; 0, 1\right) + \sum_{j:Y_{ij}=0} \log N\left(X_{ij}; AZ_i + \mu, W\right) - \lambda X_{ij}^2 \\ + \sum_{j:Y_{ij}\neq 0} \log N\left(Y_{ij}; AZ_i + \mu, W\right) + \log\left(1 - \exp\left(-\lambda Y_{ij}^2\right)\right) \end{aligned}$$

Then we can see that we need to compute expectations for $Z_{i,j}$,$Z_{i,j}^2$,$X_{i,j}$,$X_{i,j}^2$,$Z_{i,j1}Z_{i,j2}$,$X_{i,j1}Z_{i,j2}$

2.3 E-step

To calculate these expectations, we consider the joint distribution of Z and X. The prior distribution have mean $\mu^{(p)}$ and variance $\Sigma^{(p)}$.

$$\mu^{(p)} = \begin{pmatrix} 0 \\ \mu \end{pmatrix}$$

$$\Sigma^{(p)} = \begin{pmatrix} I & A^T \\ A & AA^T + W \end{pmatrix}$$

Then we can have the posterior distribution with mean $\mu^{(pos)}$ and variance $\Sigma^{(pos)}$.

$$\begin{aligned} \mu^{(pos)} &= \left(\Sigma_c^{-1} + 2\lambda I_x\right)^{-1} \\ \Sigma^{(pos)} &= \left(\Sigma_c^{-1} + 2\lambda I_x\right)^{-1} \Sigma_c^{-1} \mu_c \end{aligned}$$

where,

$$\begin{aligned} \mu_c &= \mu_0^{(p)} + \textstyle\sum_{0+}^{(p)} \left(\Sigma_{++}^{(p)}\right)^{-1} \left(Y_{i+} - \mu_+^{(p)}\right) \\ \Sigma_c &= \Sigma_{00}^{(p)} - \textstyle\sum_{0+}^{(p)} \left(\textstyle\sum_{++}^{(p)}\right)^{-1} \textstyle\sum_{+0}^{(p)} \end{aligned}$$

2.4 M-step

To optimize the log likelihood, we can first update the A and μ by solving equations $B_j\mu_j - c_j = 0$.

$$u_j = \begin{pmatrix} A_{j1} \\ A_{j2} \\ \vdots \\ A_{jD} \\ \mu_j \end{pmatrix}$$

$$B_j = \begin{pmatrix} \frac{\sum_i E[Z_{i1}Z_{i2}]}{\sum_i E\left[Z_{i1}^2\right]} & \cdots & \sum_{\sum_i E\left[Z_{i1}^2\right]} & \\ \frac{\sum_i E[Z_{i2}Z_{i1}]}{\sum_i E\left[Z_{i2}^2\right]} & 1 & \cdots & \frac{\sum_i E[Z_{i2}]}{\sum_i E\left[Z_{i2}^2\right]} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{N}\sum_i E\left[Z_{i1}\right] & \frac{1}{n}\sum_i E\left[Z_{i2}\right] & \cdots & 1 \end{pmatrix}$$

$$c_j = \begin{pmatrix} \frac{\sum_{i:Y_{ij}=0} E[X_{ij}Z_{i1}] + \sum_{i:Y_{ij}>0} Y_{ij}E[Z_{i1}]}{\sum_i E\left[Z_{i1}^2\right]} \\ \frac{\sum_{i:Y_{ij}=0} E[X_{ij}Z_{i2}] + \sum_{i:Y_{ij}>0} Y_{ij}E[Z_{i2}]}{\sum_i E\left[Z_{i2}^2\right]} \\ \vdots \\ \frac{1}{N}\left(\sum_{i:Y_{ij}=0} E\left[X_{ij}\right] + \sum_{i:Y_{ij}>0} Y_{ij}\right) \end{pmatrix}$$

Then we update W, defining $m_{i,j} = \sum_{k=1}^{K} A_{jk}Zik + \mu_j$ we have:

$$\sigma_j^2 = \frac{1}{N}\left(\sum_{i:Y_{ij}=0}\left(E\left[X_{ij}^2\right] - 2E\left[X_{ij}m_{ij}\right] + E\left[m_{ij}^2\right]\right) + \sum_{i:Y_{ij}>0}\left(Y_{ij}^2 - 2Y_{ij}E\left[m_{ij}\right] + E\left[m_{ij}^2\right]\right)\right)$$

Finally, we update λ numerically using BFGS.

3 Simulation

In this part, I generate data follow the assumed generating model, where the parameters are N=200, K=10, D=200, λ=0.05, $A \sim Uniform(-0.5, 0.5)$, $\mu \sim N(6, 1)$, $W_{i,i} \sim Uniform(0, 1)$.

Since we have already known the ground truth parameters, we can compare with the estimated parameter. However, the factor analysis model may rotate the latent variables, we can not compare Z and $\hat{Z}$, A and $\hat{A}$. Instead, we can consider the correlation of distance matrix of latent vector Z_i. Here, we choose the Spearman correlation. For the other parameters, we can compare them directly using Spearman correlation or l2 distance.

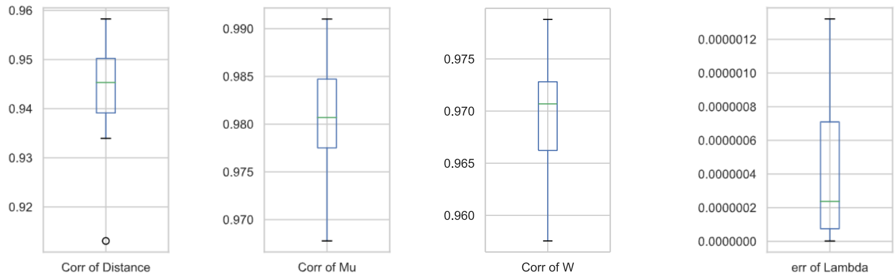


Figure 1: Compare the estimated parameters accuracy. We use the Spearman correlation of distance matrix for Z and Spearman correlation for μ and W. Squared error is used for evaluating λ. We can see from the result that the correlations are very closed to 1 and squared error is very small. So the EM algorithm for ZIFA is correct.

4 Amortized variational inference

Instead of using exact inference, we can use variational inference. Here I use variational auto-encoder with two-layer neural networks. I compared two types of VAE, the first type is original VAE with Gaussian or Poisson decoder. The second type ("Zero-inflated VAE") has two decoders: a Gaussian decoder to approximate non-zeros items and a Bernoulli encoder to simulate dropout event. This "Zero-inflated VAE" is motivated by the ZIFA model since the data is zero-inflated, learning the dropout event might benefit the latent representation. A sketch map2 shows the comparison of the VAEs.

5 Experiments on CORTEX dataset

Here we consider a real-world single cell dataset CORTEX [3]. This dataset contains 3005 cells (N = 3005) and 558 genes (D = 558), and all the cells are labeled for seven different types, for example, astrocytes ependymal, endothelial-mural.

Since we do not have ground-truth parameters for this dataset, we need to compare the clustering performance of different latent variables to measure dimensional reduction performance. To measure the quality of clustering, I use the adjusted rand index as a measure.

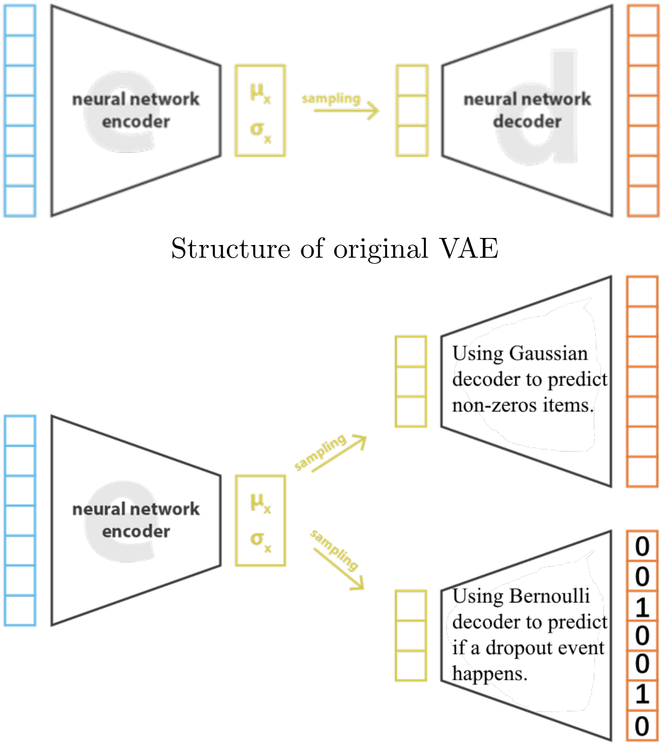
Here I choose K-Means as clustering algorithm, since K-means is sensitive to initialization, I run it 20 times with different initializations and report its adjusted rand index.

5.1 ZIFA

We need to pick the only hyper-parameter K for the ZIFA model. I searched from 2 to 20, and plot the corresponding adjusted Rand index in Figure3. Then we set K=12 for ZIFA model.

5.2 VAEs

Next, I conduct experiments on the VAEs. The hyper-parameter setting is according to experience, I list some of the important hyper-parameters. I choose 10 as the dimension


neural network encoder
μ_x
σ_x
sampling
neural network decoder
sampling
sampling
Using Gaussian decoder to predict non-zeros items.
Using Bernoulli decoder to predict if a dropout event happens.
0
0
1
0
0
1
0

Structure of original VAE

Structure of Zero-inflated VAE

Figure 2: Comparison of two types of VAE. The above one is original VAE and its decoder can be Gaussian decoder using squared loss or Poisson decoder with negative log-likelihood loss. The below one has two decoders: a Gaussian decoder to approximate non-zeros items and a Bernoulli encoder to simulate dropout events.

of latent Z, it is similar with ZIFA model. All the encoder and decoder is a two layer relu network with 1024 hidden nodes. The total epoch is 200 and batch-size is 64, so there are in total 9400 iterations. Though it is much more than ZIFA model (5 EM iterations), it is much faster than ZIFA. The Adam optimizer is used with learning rate 5e-4 and betas=0.9,0.999.

The comparison is shown in Figure 4. There are two key observations: The Poisson VAE gives the best performance, and “Zero-inflated VAE” is the worst; The variance of ZIFA K-means is significantly higher than the other VAE methods, which might suggest the latent Z extracted by ZIFA might not suitable for performing K-means clustering.

6 Conclusion and future work

In this report, I explored how to do dimensional reduction for zero-inflated data which means there are many zeros items. First, Zero Inflated Factor Analysis model is introduced and I use EM algorithm to estimate it. Simulation study and real-world experiments show its effectiveness. Then, VAE models are also adopted for this problem. I implement two types of VAEs: original VAE with Gaussian or Poisson decoder and “Zero-inflated VAE” with a Gaussian decoder to approximate non-zeros items and a Bernoulli encoder to simulate

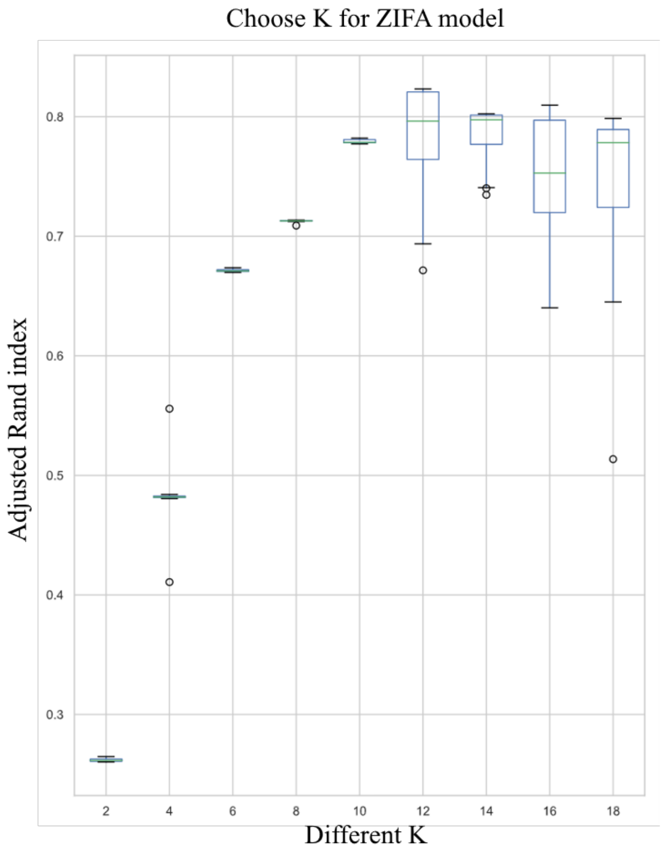


Figure 3: Choose K for ZIFA model, we can see that for small K for example 2,the adjusted Rand Index will increase when K increase, it peaks at K=12. Then the performance of clustering would be affected with an increase of latent dimension K. Also if we increase K, the algorithm runs slower. So, there we can choose K=12 for ZIFA model.

dropout events. Experiments show the VAE with Poisson decoder outperforms all the others significantly. While ZIFA also get a decent performance.

As for the modification of VAE. The “Zero-inflated VAE” model is motivated by the ZIFA model and tried to use an additional decoder to capture the dropout event. However, experiments show this does not provide any advantage to the original VAEs.

Due to the time constraint, there are still many potential improvements, the most important part is adopting VAEs to handle zero-inflated data. There are some possible reasons for the failure of “Zero-inflated VAE”, the first one is that I have limit time to tune the hyper-parameters. It might be better with an optimal setting. Next, the model structure could be improved, for example, the two decoders could share some weights or we could add one more encoder for simulating dropout event. Finally, the reconstruction loss could be improved, for example, we can design some mixture distribution to imitate the behavior of zero-inflated distribution.

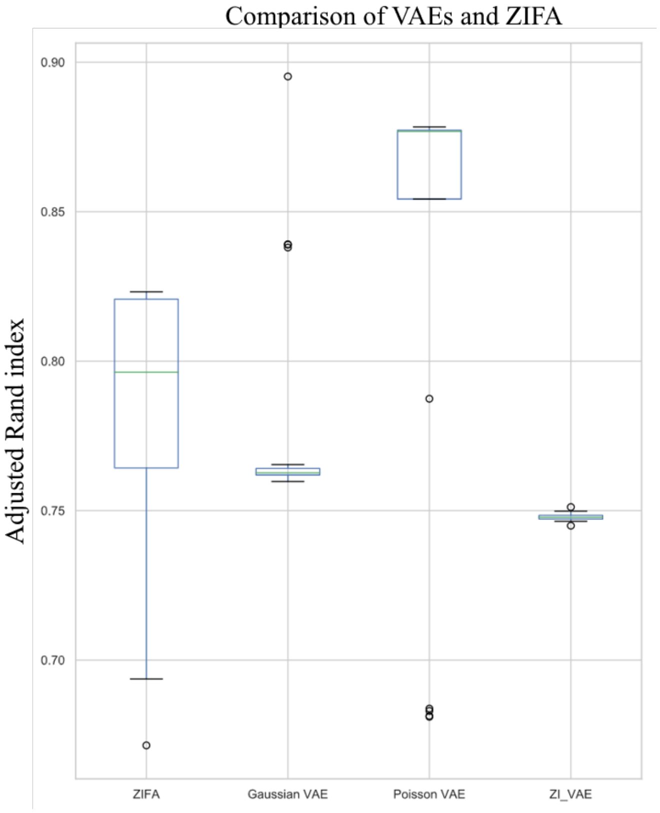


Figure 4: This figure shows comparison of VAEs and ZIFA model. All the K-means clustering is runed 20 times to reduce the randomness caused by different initialization. First we can compare the median for the four methods. The Poisson VAE outperformance all the other methods significantly. Then ZIFA follows. Guassian VAE and "Zero-inflated VAE" do not perform well comparing with others. Especially, "Zero-inflated VAE" get lowest Adjusted Rand index, it seems adding a Bernoulli encoder to simulate dropout event is not a good idea. Also, there is a difference on the variance of the metrics. We can see that ZIFA model get decent performance with quite large variance. While the others have smaller variance despite some outliers. This might suggest the hidden Z extracted from VAEs is more suitable for K-means algorithm and it is easier to distinguish different clusters.

References

[1] Diederik P Kingma and Max Welling. Auto encoding variational bayes, 2013.

[2] Emma Pierson and Christopher Yau. Zifa: Dimensionality reduction for zero-inflated single cell gene expression analysis. *bioRxiv*, 2015.

[3] Amit Zeisel, Ana B. Muñoz-Manchado, Simone Codeluppi, Peter Lönnerberg, Gioele La Manno, Anna Juréus, Sueli Marques, Hermany Munguba, Liqun He, Christer Betsholtz, Charlotte Rolny, Gonçalo Castelo-Branco, Jens Hjerling-Leffler, and Sten Lin-

narsson. Cell types in the mouse cortex and hippocampus revealed by single-cell rna-seq. *Science*, 347(6226):1138–1142, 2015.