Literature Review: Unsupervised POS Induction with Normalizing Flows

Chao Zhao

September 18, 2018

1 Generative models and normalizing flows

Generative models are a kind of approach in machine learning to calculate the probability of the input data $\mathbf{x}$. Since the modeling of $\mathbf{x}$ does not need the corresponding label $\mathbf{y}$, generative models are widely utilized in unsupervised learning. However, if $\mathbf{x}$ has high dimensions (e.g., the images and languages), it can be difficult to provide an explicit expression of its probability distribution $p_\lambda(\mathbf{x})$, and therefore the estimation of this probability would also be challenging. To address this problem, a common strategy is to first generate a latent variable $\mathbf{z}$ from a simple distribution $p_\eta(\mathbf{z})$ (e.g., an isotropic Gaussian) and then use a non-linear encoder to transform it as our observed $\mathbf{x}$. According to the Bayesian rule, the probability of $\mathbf{x}$ can be expressed as $p_\lambda(\mathbf{x}) = \int_\mathbf{z} p_\eta(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})d\mathbf{z}$. For unsupervised learning, researchers usually make $\mathbf{z}$ exactly the same as the prediction label $\mathbf{y}$.

Again, although we obtain an expression of $p_\lambda(\mathbf{x})$, calculating or maximizing it directly by marginalizing the latent variable $\mathbf{z}$ is usually intractable. Variational inference (VI) (Jordan, Ghahramani, Jaakkola, & Saul, 1999) is an effective technique to handle this problem. VI tries to maximize $p_\lambda(\mathbf{x})$ indirectly by optimizing a lower bound $\mathcal{L}_{\theta,\phi}(\mathbf{x})$ of $p_\lambda(\mathbf{x})$, based on the variational principle:

$$\begin{aligned} \log p_\lambda(\mathbf{x}) &= \mathcal{L}_{\theta,\phi}(\mathbf{x}) + \mathbb{KL}[q_\phi(\mathbf{z} \mid \mathbf{x})||p_\theta(\mathbf{z} \mid \mathbf{x})], \\ \mathcal{L}_{\theta,\phi}(\mathbf{x}) &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathbb{KL}[q_\phi(z|x)||p_\eta(\mathbf{z})], \end{aligned} \tag{1}$$

where $q_\phi(\mathbf{z} \mid \mathbf{x})$ is a posterior we introduced to approximate the real posterior $p_\theta(\mathbf{z} \mid \mathbf{x})$. By selecting some particular families of q_ϕ (e.g., the mean-field assumption), the optimization of $\mathcal{L}_{\theta,\phi}(\mathbf{x})$ can become tractable using gradient ascendant. It would, however, also impact the flexibility of q_ϕ, which makes the q_ϕ never be the same with the real posterior p_θ, i.e. the $\mathbb{KL}[q_\phi(\mathbf{z} \mid \mathbf{x})||p_\theta(\mathbf{z} \mid \mathbf{x})]$ would always be larger than 0. Many proposals have been explored to make a trade-off between the computing tractability and the model flexibility. Normalizing flow (Rezende & Mohamed, 2015) is a scalable and effective approach among these methods.

The basic idea of normalizing flow is to transform the latent variable $\mathbf{z}$ as $\mathbf{x}$ through a complex invertible function $f(\cdot)$, which is also called as a *bijector*. Following the *change of variable rule*, the probability of the $\mathbf{x}$ can be expressed using the original $p_\eta(\mathbf{z})$ with the associated Jacobian determinant of $f(\cdot)$:

$$p_\lambda(\mathbf{x}) = p_\eta(\mathbf{z})|\det \frac{\partial f^{-1}}{\partial \mathbf{x}}| = p_\eta(\mathbf{z})|\det \frac{\partial f}{\partial \mathbf{z}}|^{-1}. \tag{2}$$

By successively applying the transformations we obtain an arbitrary complex transformation $f = f_K \circ \cdots \circ f_1$, where

$$\log p_\lambda(\mathbf{x}) = \log p_\eta(\mathbf{z}) - \sum_k \log |\det \frac{\partial f_k}{\partial z_k}|, \tag{3}$$

we can find a proper f by maximizing this log-likelihood. While it is straight-forward to construct an invertible complex function using neural networks, computing the Jacobian determinant of such an arbitrary function is not trivial (need the time complexity of $O(n^3)$). Researchers therefore focus more on the design of bijectors to make the computation of the Jacobian determinant easy and without damaging its flexibility (Dinh, Krueger, & Bengio, 2014; Dinh, Sohl-Dickstein, & Bengio, 2016; Rezende & Mohamed, 2015).

2 An application in natural language processing

Generative models have achieved great performance in computer vision. But their applications in the field of natural language processing are rare because the gradient cannot be back-propagated through the discrete inputs (the tokens). Recently, He, Neubig, and Berg-Kirkpatrick (2018) propose a model that utilizes the normalizing flow for the problem of unsupervised syntactic structure prediction. They avoid the discrete input by adding a continuous word embedding layer as another latent variable. Without loss of generality, we take the part-of-speech (POS) induction task as an example to introduce their proposed method.

Unsupervised POS induction aims to assign each word w_i of a sentence $\mathbf{w}_{<1,\cdots,n>}$ with a POS tag t_i without the help of labeled data for supervised training. Traditional research adopted the hidden Markov model (HMM) (Merialdo, 1994), which is composed of a series of Markov transitions $p_\eta(t_i|t_{i-1})$ with the multinomial emissions $p_\theta(w_i|t_i)$, as the generative model for this task. The joint probability of $\mathbf{x}$ and $\mathbf{t}$ can be written as

$$p(\mathbf{x}, \mathbf{t}) = p(\mathbf{w}, \mathbf{t}) = \sum_i p_\eta(t_i|t_{i-1}) \cdot p_\theta(w_i|t_i). \tag{4}$$

However, the output tokens are discrete variables and therefore the syntactic dependencies (or similarities) among the words are lost. To depict such similarities, researchers tend to represent each word w_i as a vector (or *embedding*) $\boldsymbol{e}_i \in \mathbb{R}^d$ in a continuous low-dimensional vector space. To inform this representation into the HMM model, Lin, Ammar, Dyer, and Levin (2015) propose the Gaussian HMM, which inherits the hidden Markov chain of the POS tags from HMM, but changes the emission outputs from discrete tokens $\mathbf{w}$ to continuous token embeddings $\mathbf{e}$. To make it works, the emission probability is replaced by a multivariate Gaussian distribution. Then the joint probability becomes

$$p(\mathbf{x}, \mathbf{t}) = p(\mathbf{e}, \mathbf{t}) = \sum_i p_\eta(t_i|t_{i-1}) \cdot p_\theta(\boldsymbol{e_i}|t_i). \tag{5}$$

Again, the widely used skip-gram word embeddings (Mikolov, Chen, Corrado, & Dean, 2013) depict more on semantic similarities between words, making the embeddings of different syntactic parts overlapped and therefore not ideal for the POS induction task. To address this problem, He et al. (2018) modify the generative model by adding a latent word embedding layer $\mathbf{z}$ between the discrete POS tags $\mathbf{t}$ and the observed pre-trained word embeddings $\mathbf{e}$. They generate the latent embeddings using the same Gaussian HMM model, then transform them into the observed space using a normalizing flow $f(\cdot)$. Therefore the joint probability becomes

$$p(\mathbf{x}, \mathbf{t}) = p(\mathbf{e}, \mathbf{t}) = \sum_i p_\eta(t_i|t_{i-1}) \cdot p_\theta(\boldsymbol{z_i}|t_i) \cdot p_\phi(\boldsymbol{e_i}|\boldsymbol{z_i}). \tag{6}$$

where $p_\phi(\boldsymbol{e_i}|\boldsymbol{z_i}) = \delta(\boldsymbol{e_i} - f(\boldsymbol{z_i}))$ is a Dirac delta function to force the latent variable $\boldsymbol{z_i}$ to be mapped exactly the same as the observed embedding $\boldsymbol{e_i}$. By introducing $\boldsymbol{e}'_i = f(\boldsymbol{z}_i)$ and applying the change of variable rule discussed above, they obtain an analytical expression of the marginal emission distribution as

$$\begin{aligned} p(\boldsymbol{e}_i|t_i) &= \int_{\boldsymbol{e}'_i} p_\theta(f^{-1}(\boldsymbol{e}'_i)|t_i)\delta(\boldsymbol{e}_i - \boldsymbol{e}'_i) \left|\det \frac{\partial f^{-1}}{\partial \boldsymbol{e}'_i}\right| d\boldsymbol{e}'_i \\ &= p_\theta(f^{-1}(\boldsymbol{e}_i)|t_i) \left|\det \frac{\partial f^{-1}}{\partial \boldsymbol{e}_i}\right|. \end{aligned} \tag{7}$$

It means that by reversely projecting the observed embeddings $\mathbf{e}$ into the latent manifold $\mathbf{z}$ via f^{-1}, the model will be identical to the original Gaussian HMM with an extra determinant term as regularization. However, compared with the general embeddings, the latent embeddings would be more appropriate for separating the POS tags.

3 Conclusion

In summary, normalizing flows can explicitly generate a family of flexible distributions for variational approximation and likelihood maximization. They transform the distribution of the latent variables into that of the observed data through an invertible complex transformation flow. By deliberately designing the transformation function, the forward and the inverse computations on the flow can be implemented effectively. Normalizing flows and such generative models have achieved encouraging performance in computer vision and would also have potential applications for natural language modeling and generation.

References

Dinh, L., Krueger, D., & Bengio, Y. (2014). NICE: Non-linear Independent Components Estimation. , *1*(2), 1–13. Retrieved from http://arxiv.org/abs/1410.8516 doi: 1410.8516

Dinh, L., Sohl-Dickstein, J., & Bengio, S. (2016). Density estimation using Real NVP. Retrieved from http://arxiv.org/abs/1605.08803 doi: 1605.08803

He, J., Neubig, G., & Berg-Kirkpatrick, T. (2018). Unsupervised Learning of Syntactic Structure with Invertible Neural Projections. Retrieved from http://arxiv.org/abs/1808.09111

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., & Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, *37*(2), 183–233.

Lin, C.-C., Ammar, W., Dyer, C., & Levin, L. (2015). Unsupervised POS Induction with Word Embeddings. , 1311–1316. Retrieved from http://arxiv.org/abs/1503.06760

Merialdo, B. (1994). Tagging english text with a probabilistic model. *Computational linguistics*, *20*(2), 155–171.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Rezende, D. J., & Mohamed, S. (2015). Variational Inference with Normalizing Flows. *NIPS Workshop*, *37*, 3–6. Retrieved from http://arxiv.org/abs/1505.05770