Approximating Gradients of Expectations and It's Applications

Deniz Koyuncu

December 4, 2019

Overview

- First Half: Approximating Gradients of Expectations
  - A Review, Approximating Expectations
  - A General Objective Function
  - VI Objective Function
  - Score Functions
  - Pathwise Gradient Estimators
- Second Half: Application to VAE
  - Models with Unobserved Variables
  - EM Algorithm
  - VAE Model Description
  - Connection to EM

Model Fitting with Unobserved Variables

- There are two possible scenarios (or a combination), one encounters unobserved variables during model fitting
  1. Missing Features
  2. Hidden Variables
- In our discussion we will be focusing on the latter.

Some Examples

- Mixture of Normals

$$\begin{aligned} Z &\sim Cat(\pi) \\ X \mid Z = k &\sim N(\mu_k, \sigma_k) \end{aligned} \tag{1}$$

- HMM with categorical X

$$\begin{aligned} Z_1 &\sim Cat(\pi_1) \\ X_t \mid Z_t = k &\sim Cat(\pi_k^{(x)}) \\ Z_t \mid Z_{t-1} = k &\sim Cat(\pi_k^{(z)}) \end{aligned} \tag{2}$$

- Probabilistic PCA

$$\begin{aligned} \mathbf{Z} &\sim N(\mathbf{0}, \mathbf{I}) \\ \mathbf{X} \mid \mathbf{Z} &\sim N(\mathbf{WZ} + \boldsymbol{\mu}, \sigma\mathbf{I}) \end{aligned} \tag{3}$$

Model Fitting

- Can one still do MLE with those models?
- Let $\{\mathbf{X}_i\}_{i=1}^N$ i.i.d from $\mathbf{X}$, the log likelihood of the $\boldsymbol{\theta}$ is given as follows

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{N} \ell_i(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log P(\mathbf{X}_i; \boldsymbol{\theta}) \tag{4}$$

- $P(\mathbf{X}; \boldsymbol{\theta})$ is not defined only $P(\mathbf{X} \mid \mathbf{Z}; \boldsymbol{\theta}_x)$ and $P(\mathbf{Z}; \boldsymbol{\theta}_z)$ are defined.

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log \int P(\mathbf{X} \mid \mathbf{Z} = \mathbf{z}; \boldsymbol{\theta}_x) P(\mathbf{Z} = \mathbf{z}; \boldsymbol{\theta}_z) d\mathbf{z}$$

- Even for exponential families the above function is not generally concave.

How to Optimize?

- One can attempt to directly apply an optimization procedure to $\ell(\boldsymbol{\theta})$,
- Another approach would be using the EM algorithm,
- One formulation is of EM is as follows [2],

$$\begin{aligned}
\ell_i(\boldsymbol{\theta}) &= \log \int P(\mathbf{X}_i, \mathbf{Z}_i = \mathbf{z}; \boldsymbol{\theta}) d\mathbf{z} \\
&= \log \int q_i(\mathbf{Z}_i = \mathbf{z}) \frac{P(\mathbf{X}_i, \mathbf{Z}_i = \mathbf{z}; \boldsymbol{\theta})}{q_i(\mathbf{Z}_i = \mathbf{z})} d\mathbf{z} \\
&\geq \int q_i(\mathbf{Z}_i = \mathbf{z}) \log \frac{P(\mathbf{X}_i, \mathbf{Z}_i = \mathbf{z}; \boldsymbol{\theta})}{q_i(\mathbf{Z}_i = \mathbf{z})} d\mathbf{z} \\
&= L_i(\boldsymbol{\theta}, q_i)
\end{aligned} \tag{5}$$

- Last step comes from $-\log$ being a convex function.

EM Algorithm

$$\ell(\boldsymbol{\theta}) \geq \sum_{i=1}^{N} \int q_i(\mathbf{Z}_i = \mathbf{z}) \log \frac{P(\mathbf{X}_i, \mathbf{Z}_i = \mathbf{z}; \boldsymbol{\theta})}{q_i(\mathbf{Z}_i = \mathbf{z})} d\mathbf{z} \tag{6}$$

- Integral is outside of the log, only $P(\mathbf{X}_i, \mathbf{Z}_i = \mathbf{z}; \boldsymbol{\theta})$ depends on the parameters.
- The gap of for each data point is given by,

$$\ell_i(\boldsymbol{\theta}) - L_i(\boldsymbol{\theta}, q_i) = KL(q_i(\mathbf{Z}_i) \parallel P(\mathbf{Z}_i \mid \mathbf{X}_i; \boldsymbol{\theta})) \tag{7}$$

- If $q_i(\mathbf{Z}_i) = P(\mathbf{Z}_i \mid \mathbf{X}_i; \boldsymbol{\theta})$, than the gap would be 0, but notice that than q_i would depend on $\boldsymbol{\theta}$, we lose the simplification.

EM Algorithm Updates

- Initialize $\boldsymbol{\theta}^{(0)}$

$$
\begin{aligned}
q_i^{(t)}(\mathbf{Z}_i) &= \arg\max_{q_i} -KL(q_i(\mathbf{Z}_i) \parallel P(\mathbf{Z}_i \mid \mathbf{X}_i; \boldsymbol{\theta}^{(t-1)})) \\
&= P(\mathbf{Z}_i \mid \mathbf{X}_i; \boldsymbol{\theta}^{(t-1)}) \\
\boldsymbol{\theta}^{(t)} &= \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{N} L_i(\boldsymbol{\theta}, q_i) \\
&= \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{N} \int q_i^{(t)}(\mathbf{Z}_i = \mathbf{z}) \log P(\mathbf{X}_i, \mathbf{Z}_i = \mathbf{z}; \boldsymbol{\theta}) d\mathbf{z} \\
&= \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{N} \mathrm{E}_{q_i^{(t)}}[\log P(\mathbf{X}_i, \mathbf{Z}_i = \mathbf{z}; \boldsymbol{\theta})]
\end{aligned}
\tag{8}
$$

- The expectation step refers to finding the $q_i^{(t)}$ and calculating the expectation at the last step.
- Maximization step refers to maximizing the expectation with respect to $\boldsymbol{\theta}$.

Comments

- Under some assumptions, EM algorithm is guaranteed to converge to a critical point of $\ell(\boldsymbol{\theta})$,
- Can be a local max, a global max or a saddle point.
- Log likelihood monotonically increases i.e $\ell(\boldsymbol{\theta}^{(t)}) \geq \ell(\boldsymbol{\theta}^{(t-1)})$.
  - This is because $q_i^{(t)}(\mathbf{Z}_i) = P(\mathbf{Z}_i \mid \mathbf{X}_i; \boldsymbol{\theta}^{(t-1)})$.

EM as Coordinate Ascent

- Let us define the overall lower bound as follows,

$$\begin{aligned} \ell(\boldsymbol{\theta}) &\geq \sum_{i=1}^{N} L_i(\boldsymbol{\theta}, q_i) \\ &= L(\boldsymbol{\theta}, q_1, \ldots, q_N) \end{aligned} \tag{9}$$

- EM algorithm can be seen as a coordinate ascent optimization of the lower bound. [3]

$$\begin{aligned} q_1^{(t)}, \ldots, q_N^{(t)} &= \arg\max_{q_1, \ldots, q_N} L(\boldsymbol{\theta}^{(t-1)}, q_1, \ldots, q_N) \\ \boldsymbol{\theta}^{(t)} &= \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, q_1^{(t)}, \ldots, q_N^{(t)}) \end{aligned} \tag{10}$$

Variational Auto Encoder(VAE) Model Definition

- The VAE model[1] is defined as follows,

$$\begin{aligned} \mathbf{Z} &\sim N(\mathbf{0}, \mathbf{I}) \\ \mathbf{X} \mid \mathbf{Z} &\sim N(\vec{f_1}(\mathbf{Z}; \boldsymbol{\theta}), \vec{f_2}(\mathbf{Z}; \boldsymbol{\theta})) \end{aligned} \tag{11}$$

- Where the $\vec{f_1}$ and $\vec{f_2}$ are outputs of a Neural Network parametrized with $\boldsymbol{\theta}$.
- Notice the similarity between VAE model and the probabilistic PCA, where in the latter the $\vec{f_1}$ is a matrix multiplication with $\mathbf{Z}$.

Attempting to Use EM

- Since it's a model with unobserved variables, we can give the EM algorithm a try.
- In the update equation, we need to calculate the posterior distribution $q_i^{(t)}(\mathbf{Z}_i) = P(\mathbf{Z}_i \mid \mathbf{X}_i; \boldsymbol{\theta}^{(t-1)})$,
- For this model the integral doesn't have a closed form,

Intractable Posterior

- For simplicity assume 1D Z and X, with $\sigma = \vec{f_2} = 1$

$$
\begin{aligned}
P(Z \mid X; \boldsymbol{\theta}) &\propto P(X \mid Z; \boldsymbol{\theta}) P(Z) \\
&= \exp[-\frac{(x - f_1(z; \boldsymbol{\theta}))^2}{2}] \exp[-\frac{z^2}{2}] \\
&\propto \exp[-\frac{z^2 - 2x f_1(z; \boldsymbol{\theta}) + f_1(z; \boldsymbol{\theta})^2}{2}]
\end{aligned}
\tag{12}
$$

- It's not in the Normal distribution form or any other known distribution, because of $f_1(z; \boldsymbol{\theta})$ being a NN.

Approximating Posterior

- Remember that the reason $\ell(\boldsymbol{\theta}^{(t)}) \geq \ell(\boldsymbol{\theta}^{(t-1)})$ holds were because of using exact posterior for q.
- If we use EM algorithm with another q, it's no longer guaranteed.
- One possible approach is applying VI inference to the update equation and accepting to continue with a local minima.

$$q_i^{(t)}(\mathbf{Z}_i) = \arg\min_{q_i} KL(q_i(\mathbf{Z}_i) \parallel P(\mathbf{Z}_i \mid \mathbf{X}_i; \boldsymbol{\theta}^{(t-1)}))$$

- Mean Field closed form update equations are not going to work because of the $\vec{f_1}(\mathbf{Z}; \boldsymbol{\theta})$ term.
- One can use gradient based VI with either score function or patwise gradient estimators.

Maximization Step

- Assume we have managed to approximate the posterior distribution, the next step is

$$\boldsymbol{\theta}^{(t)} = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{N} \mathrm{E}_{q_i^{(t)}}[\log P(\mathbf{X}_i, \mathbf{Z}_i = \mathbf{z}; \boldsymbol{\theta})]$$

- The above expectation doesn't have a closed form because of $\vec{f_1}(\mathbf{Z}; \boldsymbol{\theta})$ term therefore there is no closed form solution for $\boldsymbol{\theta}^{(t)}$.
- One approach is using generalized EM which means taking a few gradient steps in each maximization step.

Maximization Step II

- We're not finding a closed form update, therefore the gradient of the expectation can be empirically estimated.

$$\boldsymbol{\nu} = \nabla_{\boldsymbol{\theta}} \sum_{i=1}^{N} \mathrm{E}_{q_i^{(t)}}[\log P(\mathbf{X}_i, \mathbf{Z}_i = \mathbf{z}; \boldsymbol{\theta})]$$

$$\hat{\boldsymbol{\nu}} = \sum_{i=1}^{N} \frac{1}{S} \sum_{s=1}^{S} \nabla_{\boldsymbol{\theta}} \log P(\mathbf{X}_i, \mathbf{Z}_i = \mathbf{z}_i^{(s)}; \boldsymbol{\theta}) \tag{13}$$

$$\text{where} \quad \mathbf{z}_i^{(s)} \sim q_i^{(t)}$$

- $\hat{\boldsymbol{\nu}}$ is an unbiased estimator of $\boldsymbol{\nu}$, we didn't need to do any trick because the parameters are not in the distribution of expectation.

Maximization Step Algorithm

- The algorithm of the maximization step is as follows
- $\boldsymbol{\theta}_0^{(t)} = \boldsymbol{\theta}^{(t-1)}$
- for r= 1,...,R
  - $\boldsymbol{\theta}_r^{(t)} = \boldsymbol{\theta}_{r-1}^{(t)} + \alpha \hat{\boldsymbol{\nu}} \mid_{\boldsymbol{\theta}_{r-1}^{(t)}}$
- $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}_{R_max}^{(t)}$

Back to Approximating the Posterior

- At the expectation step one needs to approximate the posterior distribution for $i = 1, \ldots, N$.

$$\phi_i^{(t)} = \arg\min_{\phi_i} KL(q_i(\mathbf{Z}_i; \phi_i) \parallel P(\mathbf{Z}_i \mid \mathbf{X}_i; \theta^{(t-1)}))$$

- Because we have to use gradient based VI,

$$\eta_i = \nabla_{\phi_i} KL(q_i(\mathbf{Z}_i; \phi_i) \parallel P(\mathbf{Z}_i \mid \mathbf{X}_i; \theta^{(t-1)}))$$

- We can use score function or pathwise gradient estimator $\hat{\eta}_i$.

Algorithm for Different q for Each i

- We assume the gradient based VI converges in P steps.
- for i=1,...,N
  - Initialize $\phi_{i,0}$
  - for p = 1,...,P
    - $\phi_{i,p} = \phi_{i,p-1} + \beta \hat{\eta}_i \mid_{\phi_{i,p-1}}$
  - $\phi_i^{(t)} = \phi_{i,P}$
  - $q_i^{(t)}(\mathbf{Z}_i) = q_i^{(t)}(\mathbf{Z}_i; \phi_i^{(t)})$
- If the dataset is large than the complexity becomes $N \times P$.
- There should be an alternative.

Approximating the Posterior Using a Conditional Distribution

- One possible solution is sharing the parameters of different distributions $q_i(\mathbf{Z}; \phi_i)$.
- If we use the same distribution in all of them, then it cannot approximate all of them equally well. i.e

$$q_i(\mathbf{Z}; \phi_i) = q(\mathbf{Z}; \phi)$$

- Another approach is using $q_i(\mathbf{Z}; \phi_i) = q(\mathbf{Z} \mid \mathbf{X_i}; \phi)$,
- Instead of approximating $P(\mathbf{Z} \mid \mathbf{X}; \theta)$ at each data point we're approximating for all data points.
- And at each step the overall KL minimization becomes

$$\phi^t = \arg\min_{\phi} \sum_{i=1}^{N} KL(q(\mathbf{Z_i} \mid \mathbf{X_i}; \phi) \parallel P(\mathbf{Z}_i \mid \mathbf{X}_i; \theta^{(t-1)})) \tag{14}$$

Algorithm for Conditional q

- At each expectation step the algorithm is as follows,
- Initialize ϕ_0
- for p = 1,...,P
  - $\phi_p = \phi_{p-1} + \beta \sum_{i=1}^{N} \hat{\eta}_i \mid_{\phi_{p-1}}$
- $\phi^{(t)} = \phi_P$
- $q_i^{(t)}(\mathbf{Z}_i) = q(\mathbf{Z_i} \mid \mathbf{X_i}; \phi^{(t)})$
- An observation, at each expectation step we initialize ϕ_0 and start from scratch unlike the maximiation step.

Overall Algorithm

- Expectation Step
  - Initialize ϕ_0
  - for p = 1,...,P
    - $\phi_p = \phi_{p-1} + \beta \sum_{i=1}^{N} \hat{\eta}_i \mid_{\phi_{p-1}}$
  - $\phi^{(t)} = \phi_P$
  - $q_i^{(t)}(\mathbf{Z}_i) = q(\mathbf{Z_i} \mid \mathbf{X_i}; \phi^{(t)})$
- Maximization Step
  - $\theta_0^{(t)} = \theta^{(t-1)}$
  - for r= 1,...,R
    - $\theta_r^{(t)} = \theta_{r-1}^{(t)} + \alpha \hat{\nu} \mid_{\theta_{r-1}^{(t)}}$
  - $\theta^{(t)} = \theta_R^{(t)}$

Comments

- In VAE, they use

$$q(\mathbf{Z} \mid \mathbf{X}; \phi) = N(\mathbf{Z} \mid \vec{g}_1(\mathbf{X}; \phi), \vec{g}_2(\mathbf{X}; \phi)) \tag{15}$$

- where $\vec{g}_1, \vec{g}_2$ are outputs of neural network.
- And instead of initializing ϕ_0 at each turn of expectation they use the last previous one,
- Furthermore, they select $R = 1$ and $P = 1$.
- For $\hat{\eta}$, pathwise gradient estimator is used which is called as reparametrization trick.

A Final Perspective

- EM algorithm, is a coordinate ascent algorithm

$$\begin{aligned} \boldsymbol{\phi}^{(t)} &= \arg\max_{\boldsymbol{\phi}} L(\boldsymbol{\theta}^{(t-1)}, \boldsymbol{\phi}) \\ \boldsymbol{\theta}^{(t)} &= \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \boldsymbol{\phi}^{(t)}) \end{aligned} \tag{16}$$

- In VAE, one cannot use exact coordinate ascent maximization instead ideally

$$\begin{aligned} \boldsymbol{\phi}^{(t)} &= \boldsymbol{\phi}^{(t-1)} + \beta \nabla_{\boldsymbol{\phi}} L(\boldsymbol{\theta}^{(t-1)}, \boldsymbol{\phi}) \mid_{\boldsymbol{\phi}^{(t-1)}} \\ \boldsymbol{\theta}^{(t)} &= \boldsymbol{\theta}^{(t-1)} + \alpha \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \boldsymbol{\phi}^{(t)}) \mid_{\boldsymbol{\theta}^{(t-1)}} \end{aligned} \tag{17}$$

- Where EM is guaranteed the improve the likelihood but the latter is not.


Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. 2013. arXiv: 1312.6114 [stat.ML].


Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012. ISBN: 0262018020, 9780262018029.


Radford M. Neal and Geoffrey E. Hinton. "A View of the Em Algorithm that Justifies Incremental, Sparse, and other Variants". In: *Learning in Graphical Models*. Ed. by Michael I. Jordan. Dordrecht: Springer Netherlands, 1998, pp. 355–368. ISBN: 978-94-011-5014-9. DOI: 10.1007/978-94-011-5014-9_12. URL: https://doi.org/10.1007/978-94-011-5014-9_12.