???

2017Äê6ÔÂ1ÈÕ

1 Introduction

In our original video-caption model, a topic vector is generated from the document of each (video, document) pair, and is fed to the NN model as an additional input. We trained a 2-layer MLP to predict the topic vector from the averaged video feature vector ($\theta^* = f(v)$). Hence the topic vector, which carries semantic information, serves as a supervisor signal. We believe that with the interpretive loss ($||\theta - f(v)||_2^2$) added into the objective function of the NN model, the trained features tend to carry more semantic information.

However, the topic vector carries only textual topics, but not visual topics that a video might actually include. To solve this problem and enhance the original model, a new method to improve interpretability is proposed, which allows LDA to be trained jointly with the NN model with a E-M like process. Visual topics will be developed during the joint training process.

2 Definition

In this section, we use v to denote a (averaged) video feature vector, and θ to denote the corresponding topic vector. Basic definitions of our new method are presented as follow.

Correlation Matrix. We associate v and θ with a matrix W such that $||\theta - Wv||_2$ is minimized. Hence matrix W will show the correlationship

between neuron activations in v and topict possibilities in θ. I consider Correlation Matrix being a suitable name for matrix W. Note that when θ and v are column vectors, each row of matrix W denotes the relationship between a specific topic and all video features, for

$$\theta_i = w_i v = \sum_{j=1}^{\#(features)} w_{i,j} v_j$$

Interpretive Loss. The ideal situation of W is when one topic is only related to a limited number of features. In this case, W should be a sparse matrix, hence we believe that we can minimize $||W||_0$ (number of non-zero elements in W) to enhance interpretability. Then the so-called interpretive loss should take the form

$$L_{int} = \alpha||\theta - Wv||_2^2 + ||W||_0$$

Where α is a weighting parameter. The first term proves the meaning of W, and the second term serves as the measurement of the interpretibility. Compared to the original model, we propose a direct measurement of the interpretibility.

For computational easiness, we will use $||W||_1$ (the sum of elements in W) instead of $||W||_0$. Hence the interpretive loss will be

$$L_{int} = \alpha||\theta - Wv||_2^2 + ||W||_1$$

As gradient of $||W||_1$ can be directly calculated, we can simply add this interpretive loss to the loss function of the Neural Network model and jointly optimize those two losses. Note that the l_1 formulation is not differentiable at 0, therefore a convex term should be applied to take place of the l_1 norm when the NN model is trained.

$$L_{int,NN} = \alpha||\theta - Wv||_2^2 + \sum_k \sqrt{w_k^2 + \epsilon}$$

Where ϵ is sufficiently small.

Formulated Problem. With the interpretive loss defined, we can formulize our problem to solve to be

$$minimize \;\; L_{int} = \alpha||\theta - Wv||_2^2 + ||W||_1,$$

$$s.t. \;\; L_{LDA} \leq \epsilon_1, L_{task} \leq \epsilon_2$$

Where L_{LDA} is the objective function for LDA training process, which only depends on θ. It's close form depends on which method we choose the solve the LDA model. L_{task} is the objective function of the video-caption task, and is defined as

$$L_{task} = -logp(y|x)$$

In the original model, where (x, y) is a (video, description) pair.

3 Algorithm

E-M like Algorithm. To find the best values of v, W, θ to optimize the interpretive loss, we suppose a E-M like algorithm. It consists of two steps.

- **train NN:** keep θ unchanged, and optimize v and W to minimize $\lambda_1 L_{task} + \alpha||\theta - Wv||_2^2 + ||W||_1$.

  Note that L_{LDA} is omitted, because when θ is unchanged, that part of loss is a constant value. Similar omissions will be performed later.

- **train LDA:** keep v and W unchanged and optimize θ to minimize $\lambda_2 L_{LDA} + \alpha||\theta - Wv||_2^2$. This will be done by a variational inference method.

We will iterativly execute those 2 steps to achieve our original goal, minizing L_{int}, L_{task} and L_{LDA}. The two steps are discussed further as follow.

3.1 train NN

Let the objective function of the NN model be

$$\lambda_1 L_{task} + \alpha||\theta - Wv||_2^2 + \sum_k \sqrt{w_k^2 + \epsilon}$$

And a standard gradient-based minimization procedure is performed to optimize both the task loss and the interpretive loss. This step is similar to the original model.

3.2 train LDA

This step is more complicated. Our goal is to not only optimize the LDA model to find texual topics, but also develop new topics by setting θ towards Wv. Note that when Wv is constant, we are actually using observed Wv value as a *prior* of the topic vector θ. Inspired by sLDA (Blei & McAuliffe, 2007), a supervised LDA model is proposed, and a variational inference method is applied to achieve our goal.

3.2.1 restricted LDA

The generative process of rLDA (restricted LDA) is as follows:

- Draw topic proportions $\theta|\alpha \sim Dir(\alpha)$.
- For each word
  - Draw a topic assignment $z_n|\theta \sim Multi(\theta)$.
  - Draw a word $w_n|z_n, \beta \sim Multi(\beta_{z_n})$.
- Draw a response variable $y|z_{1:N}, \delta^2 \sim N(\bar{z}, \delta^2)$, where $\bar{z} = \frac{1}{N}\sum_{i=1}^{N} z_i$.

This is a simplified sLDA, where y is sampled around $\bar{z}$. Note that $E[\bar{z}] = \theta$, therefore we can set observed value $y = Wv$, and by optimizing this rLDA model we can achieve our former goal of setting θ towards Wv.

The idea of rLDA is to feed a *prior knowledge* of θ to the LDA model as an additional input. With y $(= Wv)$ observed, rLDA tend to inference θ that is close to the prior knowledge y.

3.2.2 variational inference process of rLDA

The joint distribution of rLDA is

$$p(w, y, \theta, z|\alpha, \beta, \delta) = \prod_{d=1}^{D} Dir(\theta_d|\alpha) N(y_d|\theta_d, \delta^2) \prod_{n=1}^{N} Multi(z_{dn}|\theta_d) Multi(w_{dn}|\beta, z_{dn})$$

The variational distribution of θ, z is the same as the traditional LDA process.

$$q(\theta, z|\gamma, \phi) = Dir(\theta|\gamma) \prod_{n=1}^{N} p(z_n|\phi_n)$$

Again, we find a lower bound of the log-likelihood,

$$logp(w, y|\alpha, \beta, \delta^2) \geq L(\gamma, \phi|\alpha, \beta, \delta^2)$$

Where

$$L(\gamma, \phi|\alpha, \beta, \delta^2) = E_q[logp(w, y, \theta, z|\alpha, \beta, \delta^2)] + H[q]$$

Where

$$H[q] = E_q[-logq(\theta, z)]$$

The goal of variational inference is to maximize L, regard of β, γ, ϕ ($alpha, \delta^2$ are set as constant). The optimize process is similar to the traditional LDA model, therefore I will not give a detailed description here.

Optimize L over γ. The update formula is the same as for sLDA:

$$\gamma = \alpha + \sum_{n=1}^{N} \phi_n$$

Optimize L over ϕ. This update formula is the major difference compared to the traditional LDA. It's a simplified sLDA formula though, where $\eta = 1$. Note that

$$\frac{\partial L}{\partial \phi_{ni}} = E[log\theta_i] + E[logp(w_n|\beta)] - log\phi_{ni} + \frac{y_i}{N\delta^2} - \frac{1}{2N^2\delta^2}[2\sum_{m\neq n}^{N}\phi_{mi} + 1] + constant$$

Setting this dirivative to zero, the update formulation is shown as

$$\phi_{ni} \propto exp\{\Psi(\gamma_i) - \Psi(\sum_j \gamma_j) + log\beta_{i,w_n} + \frac{y_i}{N\delta^2} - \frac{1}{2N^2\delta^2}[2\sum_{m\neq n}^{N}\phi_{mi} + 1]\}$$

After all ϕ_{ni} values are calculated, they will be normalized to satisfy

$$\sum_i \phi_{ni} = 1$$

for every n. The first two terms in this update formulation is identical to the traditional LDA. The last two terms shows the dependency on observed value $y = Wv$. When y_i is large, ϕ_{ni} tends to be large as well.

Optimize L over β. The update equations are the same as for sLDA:

$$\beta_{k,w} \propto \sum_{d,n} \phi_{dnk} 1\{w_{dn} = w\}$$

.

constants α and δ^2 I believe that setting *alpha* and $delta^2$ to be constant is acceptable. δ^2 serves as a trade-off parameter, or it implies the importance of the observed value $y = Wv$. α is fixed as $\frac{1}{K}$ times the ones vector.

variational inference process. The overall variational inference process of rLDA with those update formulations is outlined as follow.

Input: corpus D = (y, w), constants α, δ^2
Output: Variational parameters $\{\gamma_d, \phi_d\}$, model parameters β
while *not convergence* **do**
  for $d = 1..D$ **do**
    for $n = 1..N$ **do**
      Update and normalize ϕ_{dn}.
    end
    Update γ_d.
  end
  Update β. (Could also update α and δ^2 if needed.)
end

Algorithm 1: variational inference of rLDA

4 Conclusion

This project involves two parts, the NN model and the LDA model. The NN is trained with a interpretive loss term ($l_1 - norm$ of the correlation matrix W), and the LDA is trained with topic vector θ being restricted by Wv, which can be implied as the generated topic vector by the NN model.

The first step has been tested on server, and already shows good results. The second step needs further coding.

I spent a lot of time developing my new LDA model, which turned out to be a waste of time for it has a too complicated to solve.