DD2434 Machine Learning, Advanced Large Project - Variational Inference

Jacob Hedén Malm
Ioannis Iakovidis
Marie-Ange Stefanos
Grzegorz Wozniak
Group 6

January 2022

Abstract

One of the most important and useful applications in the field of probabilistic graphical modelling is probabilistic inference, in which one uses data produced from a model with some unknown parameters to compute a probability distribution over the unknown parameters of the model. A problem with most traditional probabilistic inference methods is that a lot of graphical models are so complex and with so many inter-dependant random variables that the computational cost of probabilistic inference becomes intractable. Several approximate methods have been developed that aim to approximate these probability distributions in a more efficient manner. One of these methods is variational inference, which aims to find an approximation of the true parameter distributions by splitting the distributions into different groups, which are then assumed to be independent of each other. In this paper we try to replicate the result of David M. Blei's and Michael I. Jordan's "Variational Methods for the Dirichlet Process" paper[2], which expand the variational inference method to model mixtures created by a Dirichlet process, a distribution on distributions that allows the components of the graphical model to increase with the number of data.

1 Introduction

1.1 Description of the article

Non-parametric models are dissimilar from parametric models in that the model structure is not specified a priori but is instead determined from data. Indeed, the number and nature of the parameters are flexible and not fixed in advance. Hence, the complexity of a statistic model can directly depend in particular on the number of data points. Bayesian approach requires to define a prior probability distribution, which is the distribution actually underlying the data. One of the most famous examples one can find in the literature is the Dirichlet process (DP) whose realizations are probability distributions with a probability mass that is discrete and infinite. Since the data can be quite complex, a multimodal model can be chosen with underlying clusters or regions of different probability masses, that are latent variables since they are never observed. That is why, using a mixture model is relevant here. It is a model that encompasses several components that each has a simple parametric form. Each data point will belong to one of these components. In order to infer each of these distributions separately, we will use a Dirichlet process (DP) mixture. This model can be rephrased as the Pólya urn scheme[1] or as the Chinese restaurant process[3]. A property of the DP mixture is the absence of any assumption concerning the number of clusters or components underlying the data, that can grow as necessary. However, the DP approach has one major drawback: it relies on Monte Carlo Markov Chain (MCMC) methods for posterior inference, that is relatively slow. In the original paper[2], and thus in this project, the goal is to develop a variational inference algorithm for the DP mixture model, which is different from limited-dimensional models.

1.2 Wider context and state of art

Inference in probabilistic models is often intractable, that refers to problems for which there exist no efficient algorithms to solve them. Most sampling-based inference algorithms are instances of Markov Chain Monte-Carlo (MCMC). Two popular MCMC methods are Gibbs sampling (as described in the original paper) and Metropolis-Hastings. Nevertheless, these are sampling-based methods that show two major drawbacks. Even though they are guaranteed to find a globally optimal solution, no one can tell a priori how good this solution is and in what time duration they need to converge to it. Moreover, MCMC methods require choosing an appropriate sampling technique which is far for obvious. That is where variational techniques come in. The main idea is to cast inference as an optimization problem. Although sampling methods were historically invented first during the 1940s, variational methods have become more and more popular. Nowadays, they are the more widely used inference technique.

1.3 Objectives of the project

Since the idea is to reproduce the work done in the paper, our goals are the following. First of all, the VI algorithm presented in the algorithm required to be implemented from scratch (see equations in the Methods section), as well as generated synthetic two dimensional data according to the DP process by using The Chinese Restaurant process. Once the robot data (eight dimensional data) have been found, the idea is to replicate the VI results for both simulated and robot data. The results and comments that go with it can be found in the Results section. To conclude, the last section will discuss the methodology and approach we had during the project, explaining the differences between our results and the ones from the original paper.

2 Methods

Inference in probabilistic models is often intractable, algorithms that provide approximate solutions to the inference problem are based on subroutines that involve sampling random variables e.g. Gibbs sampling. In this report, we implement an alternative inference approach from the variational family of algorithms. Key idea of variational methods is to see inference as an optimization problem. By applying mean-field variational approach to the stick-breaking construction of the DP mixture, intractable probability distribution $p(x_n|z_n, \eta^*)$ can be factorized into variational distribution $q(v, \eta^*, z)$ (eq. 1) in which all the hidden variables are independent.

$$\begin{aligned}&q(v, \eta^*, z) = \\ &\Pi_{i=1}^{K} q(v_i|\gamma_i)\Pi_{i=1}^{K} q(\eta^* \tau_i)\Pi_{n=1}^{N} q(z_n|\phi_n)\end{aligned} \tag{1}$$

, where γ_n are the Beta parameters for the distributions on V_i, τ_i are natural parameters for the distributions on η^*, ϕ are multinomial parameters for the Z_n.

The bound on the likelihood (eq. 2) of the data is given by ELBO bound shown in [2].

$$\begin{aligned}&\log p(x|\alpha, \lambda) \geq \\ &\mathbb{E}[\log p(V|\alpha)] + \mathbb{E}[\log p(\eta^*|\lambda)] + \\ &\Sigma_{n=1}^{N} \mathbb{E}[\log p(Z_n|V)] + \\ &\mathbb{E}[\log p(x_n|z_n)] - \mathbb{E}[\log q(Z, V, \eta^*)]\end{aligned} \tag{2}$$

Next section of this paragraph shows how to calculate each part of ELBO bound.

$$\begin{aligned}&\mathbb{E}[\log p(V|\alpha)] = \\ &ln\alpha + (\alpha - 1)[\Psi(\gamma_{i2} - \Psi(\gamma_{i1} + \gamma_{i2})]\end{aligned} \tag{3}$$

$$\begin{aligned}&\mathbb{E}[\log p(\eta^*|\lambda)] = \\ &-\frac{1}{2\sigma^2}(m_{i1}^2 + s_{i1}^2) + \frac{\mu_1}{\sigma_1^2} m_{i1} - \\ &\frac{1}{2\sigma^2}(m_{i2}^2 + s_{i2}^2) + \frac{\mu_1}{\sigma_1^2} m_{i1}\end{aligned} \tag{4}$$

$$\begin{aligned}&\mathbb{E}[\log p(Z_n|V)] = \\ &\Sigma_{i=1}^{K} q(z_n > i)\mathbb{E}[\log(1 - V_i)] + \\ &q(z_n = i)\mathbb{E}[\log V_i]\end{aligned} \tag{5}$$

$$\begin{aligned}&\mathbb{E}[\log p(x_n|z_n)] = \\ &\Sigma_{j=1}^{T} \phi_{ij}[x_{i1} m_{j1} - \frac{1}{2}(m_{j1}^2 + s_{j1}^2) + \\ &x_{i2} m_{j2} - \frac{1}{2}(m_{j2}^2 + s_{j2}^2)]\end{aligned} \tag{6}$$

$$\begin{aligned}&\mathbb{E}[\log q(Z, V, \eta^*)] = \\ &\mathbb{E}[\ln q(V)] + \mathbb{E}[\ln q(Z)] + \mathbb{E}[\ln q(\eta^*)] = \\ &(\gamma_{i1} - 1)[\Psi(\gamma_{i1}) - \Psi(\gamma_{i1} + \gamma_{i2})] + \\ &(\gamma_{i2} - 1)[\Psi(\gamma_{i2}) - \Psi(\gamma_{i1} + \gamma_{i2})] + \\ &\ln T(\gamma_{i1} + \gamma_{i2}) - \ln T(\gamma_{i1}) - \ln T(\gamma_{i2}) - \\ &(\frac{1}{2} \ln s_{i1}^2 s_{i2}^2 + 1) + \Sigma_{j=1}^{T} \phi_{ij} \ln \phi_{ij}\end{aligned} \tag{7}$$

Φ is a digamma function, that comes from the derivative of the log normalization factor in the beta distribution [2]. As explained in the original paper, eq. 3, eq. 4, eq. 6, eq. 7 are derived from the standard computations from the exponential family distribution.

For each iteration of the algorithm, parameters μ, γ, ϕ for $q(\mu_t)$, $q(v_i)$, $q(z_i)$ need to be updated. Following equations show how they are computed.

$$\gamma_{i1} = 1 + \Sigma_{n=1}^{N} \phi_{ni} \tag{8}$$

$$\gamma_{i2} = \alpha + \Sigma_{n=1}^{N} \Sigma_{j=1}^{K} \phi_{nj} \tag{9}$$

$$\tau_{i1} = \lambda_1 + \Sigma_{n=1}^{N} \phi_{ni} x_n \tag{10}$$

$$\tau_{i1} = \lambda_1 + \Sigma_{n=1}^{N} \phi_{ni} \tag{11}$$

Update equations for parameters γ and τ originate from the standard computations for variational inference with exponential family distributions in a conjugate settings. Its derived by starting from the Bayes rule to define the prior and likelihood functions. When prior and likelihood gets combined, they need to be re-arranged in the exponential family distribution form so that the natural parameters can be defined. For the updates on ϕ, we use the fact that its defined for a 2-D Gaussian case (eq. 12), parameter updates for m_t and σ_t are given by eq. 13 and eq. 14

$$\begin{aligned}&\phi_i \propto \exp \Psi(\gamma_{t1}) - \Psi(\gamma_{t1} + \gamma_{t2}) + \\ &\Sigma_{j=1}^{t-1} [\Psi(\gamma_{j2}) - \Psi(\gamma_{j1} + \gamma_{j2})] + \\ &(m_{t1} + m_{t2})x_1 - [\frac{1}{2}(m_{t1}^2 + s_{t1}^2) + \frac{1}{2}(m_{t2}^2 + s_{t2}^2)]\end{aligned} \tag{12}$$

$$m_t = \frac{\Sigma_i \phi_{it} x_i + \frac{\mu_0}{\sigma_0^2}}{\Sigma_i \phi_{it} + \frac{1}{\sigma_0^2}} \tag{13}$$

$$\sigma_t = \frac{1}{\Sigma_i \phi_{it} + \frac{1}{\sigma_0^2}} \tag{14}$$

3 Results

As with the original paper, we apply our variational inference algorithm on a variety of Gaussian-Gaussian DP problems.

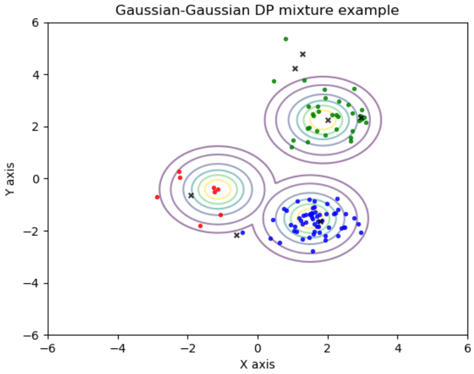

Figure 1: The approximate predictive distribution given our variational inference algorithm. The contour lines represent the inferred distributions. The color of the points represent the inferred distribution that is more likely to create them. The exes represent the means of the true distributions that created the data.

First we apply the variational inference algorithm to a simple 2-dimensional Gaussian-Gaussian mixture created by a Dirichlet process. For each of the 100 data points, we decide whether it belongs to an existing Gaussian distribution or if it belongs to a new one using the Chinese restaurant process[3]. The mean of each Gaussian distribution is sampled from a Gaussian distribution, while they all share the same diagonal covariance. As with the corresponding problem in the original paper, we truncate the number of possible distributions in our variational inference to 20. The resulting model is shown in Figure 1. As we can see, the algorithm does a good job of modelling the true Dirichlet process which created the data. It is important to note that despite the true model having 7 different distributions, our approximation uses only three, as it merges two distributions with similar means and misses some distributions that have produced a very small number of points.

In the second experiment we examine how the performance of the variational inference algorithm changes depending on the number of dimensions of the data. We create 6 datasets consisting of 200 samples, 100 for training and 100 for testing, each having a different number of dimensions, from

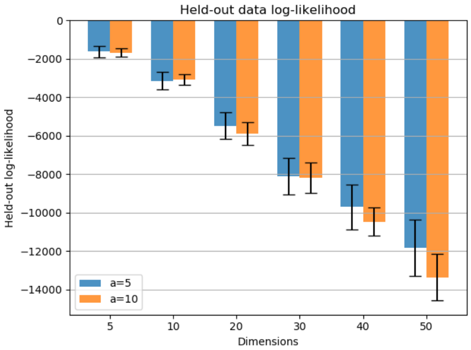

Figure 2: Log-likelihood of the held-out data (average of 20 tests).

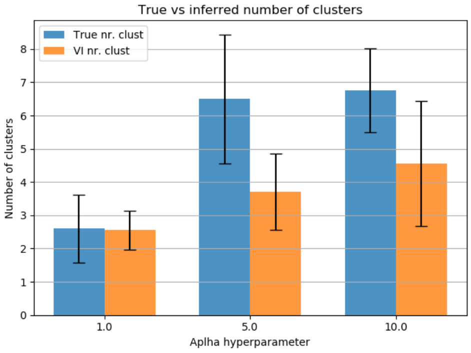

Figure 3: Number of true distributions with at least 5 data points and number of inferred distributions (average of 20 tests).

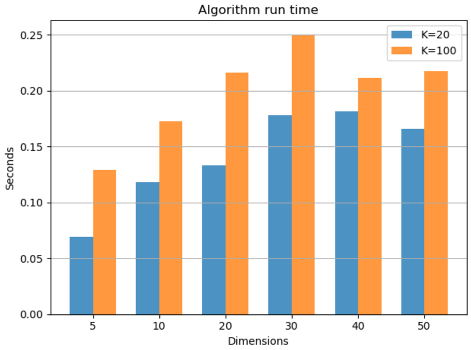

Figure 4: Run time of the variational inference algorithm (average of 20 tests).

5 to 50. Each dataset is again created using a Gaussian-Gaussian DP model, although when using the Chinese restaurant process to determine the distribution for the held-out points, each of them depends only on the distributions assigned to the 100 training points. Each Gaussian distribution has highly correlated dimensions (covariance between any two dimensions is half the variance of each dimension.). Unfortunately, the original paper provided very little information about the exact parameters of the experiment, such as the value of the hyper-parameter alpha and the prior covariance of the baseline distribution of the means. For that reason, we cannot exactly replicate their results. In spite of that, our results follow a similar patter to theirs. As expected, the more complicated the data becomes, either by having more dimensions or more distributions (higher alpha hyper-parameter), the lower the log-likelihood that our inferred distributions produce the held-out data. Moreover, we can see that increasing the value of the hyper-parameter alpha increases both the number of real and inferred distributions. Finally, as shown in Figure 4, we can see that the average run time of the variational inference algorithm increases with both the number of dimensions and the truncation parameter K. These relationships are quite logical, since higher dimension data require more computations for the same updates, and a higher K means the algorithm computes and updates parameters for more models.

The final problem involves approximating the distributions of a real world dataset. More specifically, we apply the variational inference algorithm to the 'kin-8nm' dataset, which consists of 8-dimensional samples containing the data from simulations of a robotic arm. In accordance with the original paper, we use 7000 samples for training of the variational inference algorithm and 250 for testing.

Held-out data long-likelihood (av. of 10 tests)			
K	alpha=100	alpha=1000	alpha=10000
50	−4897.69	−4778.30	−4677.4
100	−4847.93	−4673.65	−4398.47
300	−4662.35	−4406.87	−4371.22

Nr of distributions of VI model (av. of 10 tests)			
K	alpha=100	alpha=1000	alpha=10000
50	1.5	2.77777	3.39999
100	3.6	4.09999	7.1
300	3.77777	7.2	9.3

As with the previous experiment, the lack of information about the hyper-parameters used prevents us from replicating their results. However the result we get are consistent with our expectations. We see that increasing the truncation parameter of the model and increasing the hyper-parameter alpha increases the number of distributions our model uses for the data. Moreover, we see that increased values for the truncation parameter K and the value for alpha result in an increased log-likelihood for the held-out data.

4 Discussion

4.1 Comparison of found results to research paper

While we did not manage to exactly reproduce the results as given in the paper, our fitted model performed well. In the case of the robot data, tweaking of the parameters enabled us to recover a similar amount of clusters as in the paper, although the log-likelihood was consistently a little bit lower. However, when looking at the development of the log-likelihood of the validation data, clearly the pattern shows that this increases steadily, suggesting that our algorithm is actually able to learn the parameter values.

Also, when looking at the number of clusters found and fitted in the trivial 2-dimensional, N=100 case, an optical assessment tells us that our algorithm fits the data well, and for certain parameter specifications the correct cluster number was recovered.

In general, our model performed inference well.

4.2 Discussion of original methodology

A concern one might have when attempting to emulate the Dirichlet process is the existence of a truncation parameter. The goal of using a Dirichlet process prior is that we want to capture the belief we have of there being a possible infinite number of clusters, even though only a subset of these clusters have been observed, and thus given rise to data points. However, we want the model to retain the flexibility of "discovering" new clusters, when the appropriate data points are observed that might explain them. Using a truncated Dirichlet process, is of course necessary, since we cannot compute an infinite number of parameters with finite time and memory, but a natural question might be, does this limitation significantly restrict our models from the theoretical premises they are building off? Studying the properties of the Chinese restaurant process, we can conclude that cluster number k does indeed asymptotically tend to infinity. However, it is also, on average, bounded by the logarithm of the amount of data points observed.That is $\mathbb{E}[k_n \mid \alpha] = \mathcal{O}(\alpha log(n))$[3]. This means that for most practical applications, using a sufficiently large truncation parameter is feasible. For example, when generating toy-data, depending on the choice of alpha parameter, we tended to create somewhere in the range of 3-5 clusters, and our VI algorithm would recover the correct amount, or a slightly smaller amount of clusters. When running the VI algorithm on the robot data set where N=7500, and again depending on the exact values of our hyper-parameters, we tended to recover 5-7 clusters. Analysis of our VI algorithm tells us that its run-time complexity grows sublinearly with the truncation parameter K, making it feasible to choose a large truncation parameter if necessary. As such, for most non-growing data sets, the truncation parameter should not be too large a restriction, as it can be chosen to fit the size of the data.

4.3 Influence on succeeding research

The purpose of the original paper was to offer a viable alternative to inference on Dirichlet process mixture models and Bayesian non-parametric models from the previously accepted MCMC approach. This approach is accurate, however perhaps not always viable due to computational complexity, and time limitations. To test this, the authors applied two previously proven inference approaches based on MCMC to a medium sized (N=7250) 8-dimensional data set. Inference was then performed and timed, and accuracy of classification was assessed through log-likelihood of 250 held out data points from the posterior distributions. In the paper, the log-likelihood was shown to be similarly accurate as the sampling methods, while being at least 8 times faster to compute.

As such, this paper opened the door to new potential applications of Bayesian non-parametrics in general, and DP-mixtures in particular, where large amounts of data is required to be handled.

An example of this is streaming data. DP mixture models are in theory well suited to streaming type data as they are able to create new clusters "on-demand", and due to the fact that complexity of the model grows with the size of data. However, for streaming data, the truncation parameter that is necessary presents a challenge. An adaptation of the VI algorithm that removes the need for a hard truncation parameter, and successfully applies it to streaming data can be found here [5]. Similarly, further research has been done to allow this processing to occur in a distributed fashion. This was successfully accomplished, and again, applied to DPMM streaming problems [4].

These examples demonstrate the applicability of DPMM on a new domain. One that is moved into the realm of possibility by the findings in the original paper. The application of variational methods to approximate inference for Dirichlet process mixture models made them computationally feasible for new problem spaces.

References

[1] David Blackwell and James B. MacQueen. Ferguson Distributions Via Polya Urn Schemes. *The Annals of Statistics*, 1(2):353–355, March 1973. Publisher: Institute of Mathematical Statistics.

[2] David Blei and Michael Jordan. Variational methods for the dirichlet process. *Twenty-first international conference on Machine learning - ICML '04*, page 12, 09 2004.

[3] Jordan Blei. The chinese restaurant process. *Bayesian Nonparametrics*, 2007.

[4] Trevor Campbell, Julian Straub, John W Fisher III, and Jonathan P How. Streaming, distributed variational inference for bayesian nonparametrics. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Process-*

ing Systems, volume 28. Curran Associates, Inc., 2015.

[5] Viet Huynh, Dinh Phung, and Svetha Venkatesh. Streaming variational inference for dirichlet process mixtures. In Geoffrey Holmes and Tie-Yan Liu, editors, *Asian Conference on Machine Learning*, volume 45 of *Proceedings of Machine Learning Research*, pages 237–252, Hong Kong, 20–22 Nov 2016. PMLR.