Online and Distributed Learning of Gaussian Mixture Models by Bayesian Moment Matching

Priyank Jaini
Cheriton School of Computer Science
University of Waterloo
Waterloo, ON, Canada
pjaini@uwaterloo.ca

Pascal Poupart
Cheriton School of Computer Science
University of Waterloo
Waterloo, ON, Canada
ppoupart@uwaterloo.ca

Abstract

The Gaussian mixture model is a classic technique for clustering and data modeling that is used in numerous applications. With the rise of big data, there is a need for parameter estimation techniques that can handle streaming data and distribute the computation over several processors. While online variants of the Expectation Maximization (EM) algorithm exist, their data efficiency is reduced by a stochastic approximation of the E-step and it is not clear how to distribute the computation over multiple processors. We propose a Bayesian learning technique that lends itself naturally to online and distributed computation. Since the Bayesian posterior is not tractable, we project it onto a family of tractable distributions after each observation by matching a set of sufficient moments. This Bayesian moment matching technique compares favorably to online EM and online Variational Bayes in terms of time and accuracy on a set of data modeling benchmarks.

1 Introduction

Gaussian Mixture models (GMMs) [13] are simple, yet expressive distributions that are often used for soft clustering and more generally data modeling. Traditionally, the parameters of GMMs are estimated by batch Expectation Maximization (EM) [6]. However, as datasets get larger and do not fit in memory or are continuously streaming, several online variants of EM have been proposed [16, 14, 5, 11]. They process the data in one sweep by updating a sufficient statistics in constant time after each observation, however this update is approximate and stochastic, which slows down the learning rate. Furthermore it is not clear how to distribute the computation over several processors given the sequential nature of those updates.

We propose a new Bayesian learning technique that lends itself naturally to online and distributed computation. As pointed out by [4], Bayes' theorem can be applied after each observation to update the posterior in an online fashion and a dataset can be partitioned into subsets that are each processed by different processors to compute partial posteriors that can be combined into a single exact posterior that corresponds to the product of the partial posteriors divided by their respective priors.

The main issue with Bayesian learning is that the posterior may not be tractable to compute and represent. If we start with a prior that consists of the product of a Dirichlet by several Normal-Wisharts (one per Gaussian component) over the parameters of the GMM, the posterior becomes a mixture of products of Dirichlets by Normal-Wisharts where the number of mixture components grows exponentially with the number of observations. To keep the computation tractable, we project the posterior onto a single product of a Dirichlet with Normal-Wisharts by matching a set of moments of the approximate posterior with the moments of the exact posterior. While moment matching is a popular frequentist technique that can be used to estimate the parameters of a model by matching the moments of the empirical distribution of a dataset [2], here we use moment matching in a Bayesian

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

setting to project a complex posterior onto a simpler family of distributions. For instance, this type of Bayesian moment matching has been used in Expectation Propagation [12].

Despite the approximation induced by the moment matching projection, the approach compares favorably to Online EM and Variational Bayes [3] in terms of time and accuracy. Online EM requires several passes through the data before converging and therefore when it is restricted to a single pass (streaming setting), it necessarily incurs a loss in accuracy while Bayesian moment matching converges in a single pass. The approximation due to moment matching also induces a loss in accuracy, but the empirical results suggest that it is less important than the loss incurred by online EM. Finally, BMM lends itself naturally to distributed computation, which is not the case for Online EM.

2 Bayesian Moment Matching : Motivation and Algorithm

Let $\mathcal{D}_N = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ be a set of *N* data points sampled *i.i.d* from a Gaussian mixture model with M components. Assume Θ to be the parameters of this underlying Gaussian mixture model, where $\Theta = \{\theta_1, \theta_2, ..., \theta_M\}$. Each θ_i is a tuple of $(w_i, \boldsymbol{\mu}_i, \Sigma_i)\ \forall i \in \{1, 2, ..., M\}$ where w_i is the weight, $\boldsymbol{\mu}_i$ is the mean and Σ_i is the covariance matrix of the i^{th} component in the Gaussian mixture model. The aim is to find an estimate $\hat{\Theta}$ of Θ in an online (or streaming) setting given the data $\mathbf{x}^{1:N}$.

$$
\begin{aligned}
P_n(\Theta) &= Pr(\Theta|\mathbf{x}^{1:n}) \\
&\propto P_{n-1}(\Theta) Pr(\mathbf{x}_n|\Theta) \\
&\propto Pr(\Theta|\mathbf{x}^{1:n-1}) Pr(\mathbf{x}_n|\Theta) \\
&= \frac{1}{Z} Pr(\Theta|\mathbf{x}^{1:n-1}) \sum_{i=1}^{M} w_i \mathcal{N}_d\big(\mathbf{x}_n; \mu_i, \Sigma_i\big)
\end{aligned}
$$

where $Z = \int_{\Theta} Pr(\Theta|\mathbf{x}^{1:n-1}) \sum_{i=1}^{M} w_i \mathcal{N}_d\big(\mathbf{x}_n; \mu_i, \Sigma_i\big) d\Theta$

One way to estimate $\hat{\Theta}$ is to compute the posterior $P_n(\Theta) = Pr(\Theta|\mathbf{x}^{1:n})$ by using Bayes theorem recursively and the estimate being $\hat{\Theta} = \mathbb{E}[P_N(\Theta)]$. However, a major limitation with this approach is with each new data point $\mathbf{x}_j$, the number of terms in the posterior increases by a factor M due to the summation over the number of components. Hence, after *N* data points, the posterior will consist of a mixture of M^N terms, which is intractable. In this paper, we describe a Bayesian Moment Matching technique that helps to circumvent this problem.

The Bayesian Moment Matching (BMM) algorithm approximates the posterior obtained after each iteration in a manner that prevents the exponential growth of mixture terms. This is achieved by approximating the distribution $P_n(\Theta)$ obtained as the posterior by another distribution $\tilde{P}_n(\Theta)$ which is in the same family of distributions $f(\Theta|\Phi)$ as the prior by matching a set of sufficient moments *S* of $P_n(\Theta)$ with $\tilde{P}_n(\Theta)$.

Algorithm 1: Generic BMM

Input: A data set $\mathcal{D}_n := \{\mathbf{x}_i\}_{i=1}^N$
Output: $\hat{\boldsymbol{\Theta}}$: estimated parameters
Let $f(\boldsymbol{\Theta}|\boldsymbol{\Phi})$ be a family of distributions ;
Initialize a prior $P_0(\boldsymbol{\Theta}|\boldsymbol{\Phi})$;
for $n \leftarrow 1$ **to** N **do**
  Compute $P_n(\boldsymbol{\Theta})$ from $P_{n-1}(\boldsymbol{\Theta})$;
  $\forall\ g(\boldsymbol{\Theta}) \in S(f)$, evaluate $M_{g(\boldsymbol{\Theta})}(P_n)$;
  Compute $\hat{\boldsymbol{\Phi}}$ using $M_{g(\boldsymbol{\Theta})}(P_n)$'s ;
  Approximate P_n with $\tilde{P}_n(\boldsymbol{\Theta}) = f(\boldsymbol{\Theta}|\hat{\boldsymbol{\Phi}})$;
Return $\hat{\boldsymbol{\Theta}} = \mathbb{E}[\tilde{P}_N(\boldsymbol{\Theta})]$

In Algorithm 1, we describe a generic procedure to approximate the posterior P_n after each observation with a simpler distribution $\tilde{P}_n$ by moment matching. More precisely, a set of moments sufficient to define $\tilde{P}_n$ are matched with the moments of the exact posterior P_n. For every iteration, we first calculate the exact posterior $P_n(\boldsymbol{\Theta}|\mathbf{x}^{1:n})$. Then, we compute the set of moments *S(f)* that are sufficient to define a distribution in the family $f(\boldsymbol{\Theta}|\boldsymbol{\Phi})$. Next, we compute the parameter vector $\boldsymbol{\Phi}$ based on the set of sufficient moments. This determines a specific distribution $\tilde{P}_n$ in the family f that we use to approximate P_n. Note that the moments in the sufficient set *S(f)* of the approximate posterior are the same as that of the exact posterior. However, all the other moments outside this set of sufficient moments *S(f)* may not necessarily be the same. Next, we illustrate the working of the BMM algorithm for estimating the parameters of a multivariate Gaussian mixture model.

Let $\mathbf{X}^{1:n}$ be a set of *d*-dimensional *i.i.d* observations following $\Pr(\mathbf{X}|\Theta) = \sum_{i=1}^{M} w_i \mathcal{N}\big(x; \boldsymbol{\mu}_i, \Lambda^{-1}\big)$ where $\Theta = \{(w_1, \boldsymbol{\mu}_1, \Lambda_1^{-1}), (w_2, \boldsymbol{\mu}_2, \Lambda_2^{-1}), ...(w_M, \mu_M, \Lambda_M^{-1})\}$ and ***M*** is known.

We choose the prior as a product of a Dirichlet distribution over the weights $\mathbf{w}$ and M Normal-Wishart distributions corresponding to the parameters $(\boldsymbol{\mu}, \Lambda^{-1})$ of each Gaussian component. Such a prior forms a conjugate probability pair of the likelihood and is hence desirable. Concretely, $P_0(\Theta) = Dir(\mathbf{w}|\boldsymbol{\alpha}) \prod_{i=1}^{M} \mathcal{NW}(\boldsymbol{\mu}_i, \Lambda_i|\boldsymbol{\delta}_i, \kappa_i, \mathbf{W}_i, \nu_i)$ where $\mathbf{w} = (w_1, w_2, ..., w_M)$, $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_M)$, $\mathbf{W}$ is a symmetric positive definite matrix, $\kappa > 0$ is real, $\boldsymbol{\delta} \in \mathbb{R}^d$ and $\nu > d - 1$ is real. The posterior $P_1(\Theta|\mathbf{X}_1)$ after observing the first data point $\mathbf{X}_1$ is given by

$$\begin{aligned} P_1(\Theta|\mathbf{X}_1) &\propto P_0(\Theta) \Pr(\mathbf{X}_1|\Theta) \\ &\propto Dir(\mathbf{w}|\boldsymbol{\alpha}) \prod_{i=1}^{M} \mathcal{NW}(\boldsymbol{\mu}_i, \Lambda_i|\boldsymbol{\delta}_i, \kappa_i, \mathbf{W}_i, \nu_i) \sum_{j=1}^{M} w_j \mathcal{N}\big(\mathbf{X}_1; \boldsymbol{\mu}_j, \Lambda_j^{-1}\big) \end{aligned}$$

Since a Normal-Wishart distribution is a conjugate prior for a Normal distribution with unknown mean and precision matrix, $\mathcal{NW}(\boldsymbol{\mu}_i, \Lambda_i|\boldsymbol{\delta}_i, \kappa_i, \mathbf{W}_i, \nu_i)\mathcal{N}\big(\mathbf{X}_1; \boldsymbol{\mu}_i, \Lambda_i^{-1}\big) = c\mathcal{NW}(\boldsymbol{\mu}_i, \Lambda_i|\hat{\boldsymbol{\delta}}_i, \hat{\kappa}_i, \hat{\mathbf{W}}_i, \hat{\nu}_i)$ where c is some constant. Similarly, $w_j Dir(\mathbf{w}|\alpha_1, \alpha_2, .., \alpha_j, .., \alpha_M) = kDir(w_1, w_2, ..., w_M|\alpha_1, \alpha_2, ..\hat{\alpha}_j .., \alpha_M)$ where k is some constant. Therefore, $P_1(\Theta|\mathbf{X}_1)$ is

$$P_1(\Theta|\mathbf{X}_1) = \frac{1}{Z} \sum_{j=1}^{M} \left(c_j Dir(\mathbf{w}|\hat{\boldsymbol{\alpha}}_j) \mathcal{NW}(\boldsymbol{\mu}_j, \Lambda_j|\hat{\boldsymbol{\delta}}_j, \hat{\kappa}_j, \hat{\mathbf{W}}_j, \hat{\nu}_j) \prod_{i \neq j}^{M} \mathcal{NW}(\boldsymbol{\mu}_i, \Lambda_i|\boldsymbol{\delta}_i, \kappa_i, \mathbf{W}_i, \nu_i) \right)$$

where $\hat{\boldsymbol{\alpha}}_j = (\alpha_1, \alpha_2, .., \hat{\alpha}_j, .., \alpha_M)$ and Z is the normalization constant. The equation above suggests that the posterior is a mixture of product of distributions where each product component in the summation has the same form as that of the family of distributions of the prior $P_0(\Theta)$. It is evident that the terms in the posterior grow by a factor of ***M*** for each iteration, which is problematic. The Bayesian moment matching algorithm approximates this mixture $P_1(\Theta)$ with a single product of Dirichlet and Normal-Wishart distributions $\tilde{P}_1(\Theta)$ by matching all the sufficient moments of P_1 with $\tilde{P}_1$ which belongs to the same family of distributions as the prior:

$$\tilde{P}_1(\Theta) = Dir(\mathbf{w}|\boldsymbol{\alpha}^1) \prod_{i=1}^{M} \mathcal{NW}(\boldsymbol{\mu}_i, \Lambda_i|\boldsymbol{\delta}_i^1, \kappa_i^1, \mathbf{W}_i^1, \nu_i^1)$$

We evaluate the parameters $\boldsymbol{\alpha}^1, \boldsymbol{\delta}_i^1, \kappa_i^1, \mathbf{W}_i^1, \nu_i^1 \;\; \forall i \in \{1, 2, .., M\}$ by matching a set of sufficient moments of $\tilde{P}_1(\Theta)$ with $P_1(\Theta)$. The set of sufficient moments in this case is $S = \{\boldsymbol{\mu}_j, \boldsymbol{\mu}_j \boldsymbol{\mu}_j^T, \boldsymbol{\Lambda}_j, \Lambda_{j_{km}}^2, w_j, w_j^2\} \; \forall j \in 1, 2, ..., M$ where $\Lambda_{j_{km}}^2$ is the $(k, m)^{th}$ element of the matrix $\boldsymbol{\Lambda}_j$. The expressions for sufficient moments are given by $\mathbb{E}[g] = \int_\Theta g P_1(\Theta) d(\Theta)$. The parameters of $\tilde{P}_1$ can be computed from the following set of equations

$$\begin{aligned} \mathbb{E}[w_i] &= \frac{\alpha_i}{\sum_j \alpha_j}; & \mathbb{E}[w_i^2] &= \frac{(\alpha_i)(\alpha_i + 1)}{\Big(\sum_j \alpha_j\Big)\Big(1 + \sum_j \alpha_j\Big)} \\ \mathbb{E}[\boldsymbol{\Lambda}] &= \nu \mathbf{W}; & Var(\boldsymbol{\Lambda}_{ij}) &= \nu(\mathbf{W}_{ij}^2 + \mathbf{W}_{ii}\mathbf{W}_{jj}) \\ \mathbb{E}[\boldsymbol{\mu}] &= \boldsymbol{\delta}; & \mathbb{E}[(\boldsymbol{\mu} - \boldsymbol{\delta})(\boldsymbol{\mu} - \boldsymbol{\delta})^T] &= \frac{\kappa + 1}{\kappa(\nu - d - 1)} \mathbf{W}^{-1} \end{aligned}$$

Using this set of equations, the exact posterior $P_1(\Theta)$ can be approximated with $\tilde{P}_1(\Theta)$. This posterior will then be the prior for the next iteration and we keep following the steps above iteratively to finally have a distribution $\tilde{P}_n(\Theta)$ after observing a stream of data $\mathbf{X}^{1:n}$. The estimate is $\hat{\Theta} = \mathbb{E}[\tilde{P}_n(\Theta)]$.

2.1 Distributed Bayesian Moment Matching

One of the major advantages of Bayes' theorem is that the computation of the posterior can be distributed over several machines, each of which processes a subset of the data. It is also possible to compute the posterior in a distributed manner using Bayesian moment matching algorithm. For example, let us assume that we have *T* machines and a data set with *TN* data points. Each machine t,

can compute the approximate posterior $P_t(\mathbf{\Theta}|\mathbf{x}^{(t-1)N+1:tN})$ where $t \in 1, 2, .., T$ using Algorithm **??** over *N* data points. These partial posteriors $\{P_t\}_{t=1}^{T}$ can be combined to obtain a posterior over the entire data set $\mathbf{x}^{1:TN}$ according to the following equation:

$$P(\mathbf{\Theta}|\mathbf{x}^{1:TN}) = P(\mathbf{\Theta}) \prod_{t=1}^{T} \frac{P_t(\mathbf{\Theta}|\mathbf{x}^{(t-1)N+1:tN})}{P(\mathbf{\Theta})} \tag{1}$$

Subsequently, the estimate $\hat{\mathbf{\Theta}} = \mathbb{E}[P(\mathbf{\Theta}|\mathbf{x}^{1:TN})]$ is obtained over the whole data set. Therefore, we can use Bayesian moment matching algorithm to perform Bayesian learning in an online and distributed fashion.

3 Experiments

We performed experiments on 2 sets of real datasets - 10 moderate-small size datasets and 4 large datasets available publicly online at the UCI machine learning repository and Function Approximation repository[7]. All the datasets span over diverse domains. The number of attributes(or dimensions) range from 4 to 91. We subsequently compared the performance of oBMM with the online Expectation Maximization algorithm (oEM) described in [5] and online Variational Bayes (oVB) [8, 3].

We measure both - the quality of the two algorithms in terms of average log-likelihood scores on the held-out test datasets and their scalability in terms of running time. We use the Wilcoxon signed ranked test[17] to compute the *p*-value and report statistical significance with *p*-value less than 0.05, to test the statistical significance of the results. We computed the parameters for each algorithm over a range of components varying from 2 to 10. For analysis, we report the model for which the log-likelihood over the test data stabilized and showed no further significant improvement. For oEM the step size for the stochastic approximation in the E-Step was set to $(n + 3)^{-\alpha}$ where $0.5 \leq \alpha \leq 1$[11] where *n* is the number of observations. We evaluate the performance of online Distributed Moment Matching (oDMM) by dividing the training datasets in to 5 smaller data sets, and processing each of these small datasets on a different machine. The output from each machine is collected and combined to give a single estimate for the parameters of the model learned.

Table 1: Log-likelihood scores on 10 data sets. The best results among oBMM and oEM are highlighted in bold font. ↑(or ↓) indicates that the method has significantly better (or worse) log-likelihoods than Online Bayesian Moment Matching (oBMM) under Wilcoxon signed rank test with pvalue < 0.05.

DATA SET	INSTANCES	OVB	OEM	OBMM
ABALONE	4177	2.18↓	-2.65 ↓	**-1.82**
BANKNOTE	1372	9.89 ↓	-9.74 ↓	**-9.65**
AIRFOIL	1503	16.71 ↓	**-15.86**	-16.53
ARABIC	8800	15.42↓	-15.83 ↓	**-14.99**
TRANSFUSION	748	13.31 ↓	-13.26 ↓	**-13.09**
CCPP	9568	16.87↓	-16.53 ↓	**-16.51**
COMP. ACTIVITY	8192	-121.55↓	-132.04 ↓	**-118.82**
KINEMATICS	8192	10.51 ↓	-10.37 ↓	**-10.32**
NORTHRIDGE	2929	19.03↓	-18.31 ↓	**-17.97**
PLASTIC	1650	9.39 ↓	-9.47 ↓	**-9.01**

Table 1 shows the average log-likelihood on test sets for oVB, oEM and oBMM. oBMM outperforms oEM on 9 of the 10 datasets and it outperformed oVB on all datasets. The results show that for some datasets, oBMM has significantly better log-likelihoods than both oEM and oVB. Table 2 shows the log-likelihood scores and running times of each algorithm on large datasets. In terms of log-likelihood scores, oBMM outperforms oEM, oVB and oDMM on all 4 datasets. While, the performance of oDMM is expected to be worse than oBMM, it is to be noticed that the performance of oDMM is not very significantly worse. This is encouraging in light of the huge gains in terms of running time of oDMM over oVB, oEM and oBMM. Table 2 shows the performance of each algorithm in terms of running times. oDMM outperforms each of the other algorithms very significantly. It is also worth noting that oBMM performed better than oEM on 3 out of 4 datasets.

Table 2: Log-likelihood scores and Avg. running time on 4 large data sets. The best results among oBMM, oDMM and oEM are highlighted in bold font.

Data		Avg. Log-Likelihood				Avg. Running Time			
Data (Attributes)	Instances	oVB	oEM	oBMM	oDMM	oVB	oEM	oBMM	oDMM
Heterogeneity (16)	3930257	-175.3↓	-176.2↓	**-174.3**	-180.7	87.3	77.3	81.7	**17.5**
Magic 04 (10)	19000	-32.9↓	-33.4↓	**-32.1**	-35.4	8.1	7.3	6.8	**1.4**
Year MSD (91)	515345	-514.6↓	-513.7↓	**-506.5**	-513.8	473.7	336.5	108.2	**21.2**
MiniBooNe (50)	130064	-58.6↓	-58.1↓	**-54.7**	-60.3	57.6	48.6	12.1	**2.3**

4 Discussion

Expectation Maximization, Variational Bayes (VB) and Markov Chain Monte Carlo (MCMC) are popular techniques used for approximate Bayesian learning. However, MCMC is difficult to run in an online fashion. This is due to the fact that BMM is naturally online and therefore does not require mini-batches. In contrast, in order to run in an online fashion Variational Bayes requires mini-batches and a decreasing learning rate, however the size of the mini-batches and the decay procedure for the learning rate require some fine tuning. In general, the use of mini-batches always leads to some information loss since data in previous mini-batches is not accessible. BMM does not suffer from this type of information loss and there is no batch size nor learning rate to fine tune. While online variants of the Expectation Maximization (EM) algorithm exist, their data efficiency is reduced by a stochastic approximation of the E-step and it is not clear how to distribute the computation over multiple processors.

Another aim of this work was to develop a robust online and distributed algorithm for mixture models. While spectral learning [9, 1, 10] has generated a lot of interest because of its ability to produce consistent estimates (under suitable conditions) for various latent variable models, it often generates negative probabilities in practice which is problematic - a poor choice of the rank of the model parameters and insufficient training data increase the likelihood of negative probabilities [18]. We believe BMM can be demonstrated to be a consistent online algorithm for mixture models under certain conditions. The sequence of predictive posterior obtained from the Bayesian Moment Matching algorithm forms a martingale sequence when specific moments are matched. Further, in the domain of discrete distributions, the moment matching procedure in BMM can be recast as Robbins-Monro method [15] and can be shown to converge to the true parameters. We will explore the theoretical properties of BMM in future work.

5 Conclusion

In this paper, we explored online algorithms to learn the parameters of Gaussian Mixture models. We proposed an online Bayesian Moment Matching algorithm for parameter learning and demonstrated how it can be used in a distributed manner leading to substantial gains in running time. We further showed through empirical analysis that the online Bayesian Moment outperforms online EM and online VB both in terms of accuracy and running time. We also demonstrated that distributing the algorithm over several machines results in faster running times without significantly compromising accuracy, which is particularly advantageous when running time is a major bottleneck.

References

[1] Anima Anandkumar, Dean P Foster, Daniel J Hsu, Sham M Kakade, and Yi-Kai Liu. A spectral algorithm for latent dirichlet allocation. In *Advances in Neural Information Processing Systems*, pages 917–925, 2012.

[2] Animashree Anandkumar, Daniel Hsu, and Sham M. Kakade. A method of moments for mixture models and hidden markov models. *Journal of Machine Learning Research - Proceedings Track*, 23:33.1–33.34, 2012.

[3] Matthew James Beal. *Variational algorithms for approximate Bayesian inference*. University of London London, 2003.

[4] Tamara Broderick, Nicholas Boyd, Andre Wibisono, Ashia C Wilson, and Michael I Jordan. Streaming variational bayes. In *Advances in Neural Information Processing Systems*, pages 1727–1735, 2013.

[5] Olivier Cappé and Eric Moulines. On-line expectation–maximization algorithm for latent data models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(3):593–613, 2009.

[6] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[7] H. Altay Guvenir and I. Uysal. Bilkent university function approximation repository. 2000.

[8] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

[9] Daniel Hsu, Sham M Kakade, and Tong Zhang. A spectral algorithm for learning hidden markov models. *Journal of Computer and System Sciences*, 78(5):1460–1480, 2012.

[10] Ravindran Kannan, Santosh Vempala, et al. Spectral algorithms. *Foundations and Trends® in Theoretical Computer Science*, 4(3–4):157–288, 2009.

[11] Percy Liang and Dan Klein. Online em for unsupervised models. In *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics*, pages 611–619. Association for Computational Linguistics, 2009.

[12] Thomas Minka and John Lafferty. Expectation-propagation for the generative aspect model. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 352–359. Morgan Kaufmann Publishers Inc., 2002.

[13] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[14] 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*, pages 355–368. Springer, 1998.

[15] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[16] D. M. Titterington. Recursive parameter estimation using incomplete data. page 46(2):257–267, 1984.

[17] Frank Wilcoxon. Some rapid approximate statistical procedures. *Annals of the New York Academy of Sciences*, pages 808–814, 1950.

[18] Han Zhao and Pascal Poupart. A sober look at spectral learning. *arXiv preprint arXiv:1406.4631*, 2014.

Supplementary Material

Priyank Jaini
Cheriton School of Computer Science
University of Waterloo
Waterloo, ON, Canada
pjaini@uwaterloo.ca

Pascal Poupart
Cheriton School of Computer Science
University of Waterloo
Waterloo, ON, Canada
ppoupart@uwaterloo.ca

1 Background

1.1 Moment Matching

A moment is a quantitative measure of the shape of a distribution or a set of points. Let $f(\boldsymbol{\theta}|\boldsymbol{\phi})$ be a probability distribution over a d-dimensional random variable $\boldsymbol{\theta} = \{\theta_1, \theta_2, ..., \theta_d\}$. The j^{th} order moments of $\boldsymbol{\theta}$ are defined as $M_{g_j(\boldsymbol{\theta})}(f) = \mathbb{E}\Big[\prod_i \theta_i^{n_i}\Big]$ where $\sum_i n_i = j$ and g_j is a monomial of $\boldsymbol{\theta}$ of degree *j*.

$$M_{g_j(\boldsymbol{\theta})}(f) = \int_{\boldsymbol{\theta}} g_j(\boldsymbol{\theta}) f(\boldsymbol{\theta}|\boldsymbol{\phi}) d\boldsymbol{\theta}$$

For some distributions *f*, there exists a set of monomials *S(f)* such that knowing $M_g(f)\ \forall g \in S(f)$ allows us to calculate the parameters of *f*. For example, for a Gaussian distribution $\mathcal{N}(x; \mu, \sigma^2)$, the set of sufficient moments *S(f)* = $\{x, x^2\}$. This means knowing M_x and M_{x^2} allows us to estimate the parameters μ and σ^2 that characterize the distribution. We use this concept called the method of moments in our algorithm.

Method of Moments is a popular frequentist technique used to estimate the parameters of a probability distribution based on the evaluation of the empirical moments of a dataset. It has been previously used to estimate the parameters of latent Dirichlet allocation, mixture models and hidden Markov models [1]. Method of Moments or moment matching technique can also be used for a Bayesian setting by computing a subset of the moments of the intractable posterior distribution given by Eq. **??**. Subsequently, another tractable distribution from a family of distributions that matches the set of moments can be selected as an approximation for the intractable posterior distribution. For Gaussian mixture models, we use the Dirichlet as a prior over the weights of the mixture and a Normal-Wishart distribution as a prior over each Gaussian component. We next give details about the Dirichlet and Normal-Wishart distributions, including their set of sufficient moments.

1.2 Family of Prior Distributions

In Bayesian Moment Matching, we project the posterior onto a tractable family of distribution by matching a set of sufficient moments. To ensure scalability, it is desirable to start with a family of distributions that is a conjugate prior pair for a multinomial distribution (for the set of weights) and Gaussian distribution with unknown mean and covariance matrix. The product of a Dirichlet distribution over the weights with a Normal-Wishart distribution over the mean and covariance matrix of each Gaussian component ensures that the posterior is a mixture of products of Dirichlet and Normal-Wishart distributions. Subsequently, we can approximate this mixture in the posterior with a single product of Dirichlet and Normal-Wishart distributions by using moment matching. We explain this in greater detail in Section **??**, but first we describe briefly the Normal-Wishart and Dirichlet distributions along with some sets of sufficient moments.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

1.2.1 Dirichlet Distribution

The Dirichlet distribution is a family of multivariate continuous probability distributions over the interval [0,1]. It is the conjugate prior probability distribution for the multinomial distribution and hence it is a natural choice of prior over the set of weights $\mathbf{w} = \{w_1, w_2, ..., w_M\}$ of a Gaussian mixture model. A set of sufficient moments for the Dirichlet distribution is $S = \{(w_i, w_i^2) : \forall i \in \{1, 2, ..., M\}\}$. Let $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, ..., \alpha_M\}$ be the parameters of the Dirichlet distribution over $\mathbf{w}$, then

$$\begin{aligned}
\mathbb{E}[w_i] &= \frac{\alpha_i}{\sum_j \alpha_j} \quad \forall i \in \{1, 2, ..., M\} \\
\mathbb{E}[w_i^2] &= \frac{(\alpha_i)(\alpha_i + 1)}{\Big(\sum_j \alpha_j\Big)\Big(1 + \sum_j \alpha_j\Big)} \quad \forall i \in \{1, 2, ..., M\}
\end{aligned} \tag{1}$$

1.2.2 Normal Wishart Prior

The Normal-Wishart distribution is a multivariate distribution with four parameters. It is the conjugate prior of a multivariate Gaussian distribution with unknown mean and covariance matrix [3]. This makes a Normal-Wishart distribution a natural choice for the prior over the unknown mean and precision matrix for our case.

Let $\boldsymbol{\mu}$ be a d-dimensional vector and $\boldsymbol{\Lambda}$ be a symmetric positive definite $d \times d$ matrix of random variables respectively. Then, a Normal-Wishart distribution over $(\boldsymbol{\mu}, \boldsymbol{\Lambda})$ given parameters $(\boldsymbol{\mu}_0, \kappa, \mathbf{W}, \nu)$ is such that $\boldsymbol{\mu} \sim \mathcal{N}_d\big(\boldsymbol{\mu}; \boldsymbol{\mu}_0, (\kappa\boldsymbol{\Lambda})^{-1}\big)$ where $\kappa > 0$ is real, $\boldsymbol{\mu}_0 \in \mathbb{R}^d$ and $\boldsymbol{\Lambda}$ has a Wishart distribution given as $\boldsymbol{\Lambda} \sim \mathcal{W}(\boldsymbol{\Lambda}; \mathbf{W}, \nu)$ where $\mathbf{W} \in \mathbb{R}^{d \times d}$ is a positive definite matrix and $\nu > d - 1$ is real. The marginal distribution of $\boldsymbol{\mu}$ is a multivariate t-distribution i.e $\boldsymbol{\mu}|\boldsymbol{\Lambda} \sim t_{\nu-d+1}\big(\boldsymbol{\mu}; \boldsymbol{\mu}_0, \frac{\mathbf{W}}{\kappa(\nu-d+1)}\big)$. The univariate equivalent for the Normal-Wishart distribution is the Normal-Gamma distribution.

In Section 1.1, we defined *S*, a set of sufficient moments to characterize a distribution. In the case of the Normal-Wishart distribution, we would require at least four different moments to estimate the four parameters that characterize it. A set of sufficient moments in this case is $S = \{\boldsymbol{\mu}, \boldsymbol{\mu}\boldsymbol{\mu}^T, \boldsymbol{\Lambda}, \Lambda_{ij}^2\}$ where Λ_{ij}^2 is the $(i, j)^{th}$ element of the matrix $\boldsymbol{\Lambda}$. The expressions for sufficient moments are given by

$$\begin{aligned}
\mathbb{E}[\boldsymbol{\mu}] &= \boldsymbol{\mu}_0 \\
\mathbb{E}[(\boldsymbol{\mu} - \boldsymbol{\mu}_0)(\boldsymbol{\mu} - \boldsymbol{\mu}_0)^T] &= \frac{\kappa + 1}{\kappa(\nu - d - 1)}\mathbf{W}^{-1} \\
\mathbb{E}[\boldsymbol{\Lambda}] &= \nu\mathbf{W} \\
Var(\Lambda_{ij}) &= \nu(W_{ij}^2 + W_{ii}W_{jj})
\end{aligned} \tag{2}$$

1.3 Online Expectation Maximization

Batch Expectation Maximization [4] is often used in practice to learn the parameters of the underlying distribution from which the given data is assumed to be derived. In [8], a first online variant of EM was proposed, which was later modified and improved in several variants [6, 7, 2, 5] that are closer to the original batch EM algorithm. In online EM, an updated parameter estimate $\hat{\boldsymbol{\Theta}}_n$ is produced after observing each data instance $\mathbf{x}_n$. This is done by replacing the expectation step by a stochastic approximation, while the maximization step is left unchanged. In the limit, online EM converges to the same estimate as batch EM when it is allowed to do several iterations over the data. Hence, a loss in accuracy is incurred when it is restricted to a single pass over the data as required in the streaming setting.

References

[1] Animashree Anandkumar, Daniel Hsu, and Sham M. Kakade. A method of moments for mixture models and hidden markov models. *Journal of Machine Learning Research - Proceedings Track*, 23:33.1–33.34, 2012.

[2] Olivier Cappé and Eric Moulines. On-line expectation–maximization algorithm for latent data models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(3):593–613, 2009.

[3] Morris H. Degroot. *Optimal statistical décisions*. McGraw-Hill Book Company, New York, St Louis, San Francisco, 1970.

[4] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[5] Percy Liang and Dan Klein. Online em for unsupervised models. In *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics*, pages 611–619. Association for Computational Linguistics, 2009.

[6] 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*, pages 355–368. Springer, 1998.

[7] Masa-Aki Sato and Shin Ishii. On-line em algorithm for the normalized gaussian network. *Neural computation*, 12(2):407–432, 2000.

[8] D. M. Titterington. Recursive parameter estimation using incomplete data. page 46(2):257–267, 1984.