Online Learning of Nonparametric Mixture Models via Sequential Variational Approximation

Authored by:

Dahua Lin

Abstract

Reliance on computationally expensive algorithms for inference has been limiting the use of Bayesian nonparametric models in large scale applications. To tackle this problem, we propose a Bayesian learning algorithm for DP mixture models. Instead of following the conventional paradigm – random initialization plus iterative update, we take an progressive approach. Starting with a given prior, our method recursively transforms it into an approximate posterior through sequential variational approximation. In this process, new components will be incorporated on the fly when needed. The algorithm can reliably estimate a DP mixture model in one pass, making it particularly suited for applications with massive data. Experiments on both synthetic data and real datasets demonstrate remarkable improvement on efficiency – orders of magnitude speed-up compared to the state-of-the-art.

1 Paper Body

Bayesian nonparametric mixture models [7] provide an important framework to describe complex data. In this family of models, Dirichlet process mixture models (DPMM) [1, 15, 18] are among the most popular in practice. As opposed to traditional parametric models, DPMM allows the number of components to vary during inference, thus providing great flexibility for explorative analysis. Nonetheless, the use of DPMM in practical applications, especially those with massive data, has been limited due to high computational cost. MCMC sampling [12, 14] is the conventional approach to Bayesian nonparametric estimation. With heavy reliance on local updates to explore the solution space, they often show slow mixing, especially on large datasets. Whereas the use of split-merge moves and data-driven proposals [9,17,20] has substantially improved the mixing performance, MCMC methods still require many passes over a dataset to reach the equilibrium distribution. Variational inference [4, 11, 19, 22], an alternative approach based on mean field approximation, has become increasingly popular recently due to better run-time performance. Typical variational methods for nonparametric mixture models rely on a truncated approximation of the

stick breaking construction [16], which requires a fixed number of components to be maintained and iteratively updated during inference. The truncation level are usually set conservatively to ensure approximation accuracy, incurring considerable amount of unnecessary computation. The era of Big Data presents new challenges for machine learning research. Many real world applications involve massive amount of data that even cannot be accommodated entirely in the memory. Both MCMC sampling and variational inference maintain the entire configuration and perform iterative updates of multiple passes, which are often too expensive for large scale applications. This challenge motivated us to develop a new learning method for Bayesian nonparametric models that can handle massive data efficiently. In this paper, we propose an online Bayesian learning algorithm for generic DP mixture models. This algorithm does not require random initialization of components. Instead, it begins with the prior DP(??) and progressively transforms it into an approximate posterior of the mixtures, with new components introduced on the fly as needed. Based on a new way of variational approximation, the algorithm proceeds sequentially, taking in one sample at a time to make the 1

update. We also devise specific steps to prune redundant components and merge similar ones, thus further improving the performance. We tested the proposed method on synthetic data as well as two real applications: modeling image patches and clustering documents. Results show empirically that the proposed algorithm can reliably estimate a DP mixture model in a single pass over large datasets.

2

Related Work

Recent years witness lots of efforts devoted to developing efficient learning algorithms for Bayesian nonparametric models. A n important line of research is to accelerate the mixing in MCMC through better proposals. Jain and Neal [17] proposed to use split-merge moves to avoid being trapped in local modes. Dahl [6] developed the sequentially allocated sampler, where splits are proposed by sequentially allocating observations to one of two split components through sequential importance sampling. This method was recently extended for HDP [20] and BP-HMM [9]. There has also been substantial advancement in variational inference. A significant development along is line is the Stochastic Variational Inference, a framework that incorporates stochastic optimization with variational inference [8]. Wang et al. [23] extended this framework to the non-parametric realm, and developed an online learning algorithm for HDP [18]. Wang and Blei [21] also proposed a truncation-free variational inference method for generic BNP models, where a sampling step is used for updating atom assignment that allows new atoms to be created on the fly. Bryant and Sudderth [5] recently developed an online variational inference algorithm for HDP, using mini-batch to handle streaming data and split-merge moves to adapt truncation levels. They tried to tackle the problem of online BNP learning as we do, but via a different approach. First, we propose a generic method while they focuses on topic models. The designs are also different ? our method starts from scratch and progressively adds new components. Its overall complexity is O(nK), where

n and K are number of samples and expected number of components. Bryant?s method begins with random initialization and relies on splits over mini-batch to create new topics, resulting in the complexity of O(nKT), where T is the number of iterations for each mini-batch. The differences stem from the theoretical basis ? our method uses sequential approximation based on the predictive law, while theirs is an extension of the standard truncation-based model. Nott et al. [13] recently proposed a method, called VSUGS, for fast estimation of DP mixture models. Similar to our algorithm, the VSUGS method proposed takes a sequential updating approach, but relies on a different approximation. Particularly, what we approximate is a joint posterior over both data allocation and model parameters, while VSUGS is based on the approximating the posterior of data allocation. Also, VSUGS requires fixing a truncation level T in advance, which may lead to difficulties in practice (especially for large data). Our algorithm provides a way to tackle this, and no longer requires fixed truncation.

3

Nonparametric Mixture Models

This section provide a brief review of Dirichlet Process Mixture Model ? one of the most widely used nonparametric mixture models. A Dirichlet Process (DP), typically denoted by DP(??) is characterized by a concentration parameter ? and a base distribution ?. It has been shown that sample paths of a DP are almost surely discrete [16], and can be expressed as D=

? X

?k ??k ,

with ?k = vk

k=1

k?1 Y

vl , vk ? Beta(1, ?k), ?k = 1, 2, ...

(1)

l=1

This is often referred to as the stick breaking representation, and ?k is called an atom. Since an atom can be repeatedly generated from D with positive probability, the number of distinct atoms is usually less than the number of samples. The Dirichlet Process Mixture Model (DPMM) exploits this property, and uses a DP sample as the prior of component parameters. Below is a formal definition: D ? DP (??),

?i ? ?, xi ? F (?—?i), ?i = 1, . . . , n.

(2)

Consider a partition {C1 , . . . , CK } of {1, . . . , n} such that ?i are identical for all i ? Ck , which we denote by ?k . Instead of maintaining ?i explicitly, we introduce an indicator zi for each i with 2

?i = ?zi . Using this clustering notation, this formulation can be rewritten equivalently as follows: z1:n ? CRP(?), ?k ? ?, ?k = 1, 2, . . . K xi ? F (?—?zi), ?i = 1, 2, . . . , n. (3) Here, CRP(?) denotes a Chinese Restaurant Prior, which is a distribution over exchangeable partitions. Its probability mass function is given by K ?(?)?K Y pCRP (z1:n —?) = ?(—Ck —). (4) ?(? + n) k=1

4

Variational Approximation of Posterior

Generally, there are two approaches to learning a mixture model from observed data, namely Maximum likelihood estimation (MLE) and Bayesian learning. Specifically, maximum likelihood estimation seeks an optimal point estimate of ?, while Bayesian learning aims to derive the posterior distribution over the mixtures. Bayesian learning takes into account the uncertainty about ?, often resulting in better generalization performance than MLE. In this paper, we focus on Bayesian learning. In particular, for DPMM, the predictive distribution of component parameters, conditioned on a set of observed samples x1:n , is given by (5) p(?0 —x1:n) = ED—x1:n [p(?0 —D)] . Here, ED—x1:n takes the expectation w.r.t. p(D—x1:n). In this section, we derive a tractable approximation of this predictive distribution based on a detailed analysis of the posterior. 4.1

Posterior Analysis

Let D ? DP(??) and ?1 , . . . , ?n be iid samples from D, {C1 , . . . , CK } be a partition of {1, . . . , n} such that ?i for all i ? Ck are identical, and ?k = ?i ?i ? Ck . Then the posterior distribution of D remains a DP, as D—?1:n ? DP(? ?? ?), where ? ? = ? + n, and K

? ?=

X —Ck — ? ?+ ?? . ?+n ?+n k

(6)

k=1

The atoms are generally unobservable, and therefore it is more interesting in practice to consider the posterior distribution of D given the observed samples. For this purpose, we derive the lemma below that provides a constructive characterization of the posterior distribution given both the observed samples x1:n and the partition z. Lemma 1. Consider the DPMM in Eq.(3). Drawing a sample from the posterior distribution p(D—z1:n , x1:n) is equivalent to constructing a random probability measure as follows ?0 D 0 +

K X

?k ??k ,

k=1

with D0 ? DP(??), (?0 , ?1 , . . . , ?k) ? Dir(?, m1 , . . . , mK), ?k ? ?—Ck . (7) Q Here, mk = —Ck —, ?—Ck is a posterior distribution given by i.e. ?—Ck (d?) ? ?(d?) i?Ck F (xi —?). This lemma immediately follows from the Theorem 2 in [10] as DP is a special case of the socalled Normalized Random Measures with Independent Increments (NRMI). It is worth emphasizing that p(D—x, z) is no longer a Dirichlet process, as the locations of the atoms ?1 , . . . , ?K are nondeterministic, instead they follow the posterior distributions ?—Ck . By marginalizing out the partition z1:n , we obtain the posterior distribution p(D—x1:n): X p(D—x1:n) = p(z1:n —x1:n)p(D—x1:n , z1:n).

(8)

z1:n (z)

(z)

Let {C1 , . . . , CK } be the partition corresponding to z1:n , we have (z) K YZ Y p(z1:n —x1:n) ? pCRF (z1:n —?) ?(d?k) F (xi —?k). k=1

3

(z)

i?Ck

(9)

4.2

Variational Approximation

Computing the predictive distribution based on Eq.(8) requires enumerating all possible partitions, which grow exponentially as n increases. To tackle this difficulty, we resort to variational approximation, that is, to choose a tractable distribution to approximate p(D—x1:n , z1:n). In particular, we consider a family of random probability measures that can be expressed as follows: n XY q(D—?, ?) = ?i (zi)q?(z) (D—z1:n). (10) z1:n i=1

Here,

(z) q? (D—z1:n) d

is a stochastic process conditioned on z1:n , defined as

q?(z) (D—z1:n) ? ?0 D0 +

K X

?k ??k ,

k=1 (z)

(z)

with D0 ? DP(??), (?0 , ?1 , . . . , ?K) ? Dir(?, m1 , . . . , mK), ?k ? ?k . (11) d

(z)

Here, we use ? to indicate that drawing a sample from q? is equivalent to constructing one according (z) (z) to the right hand side. In addition, mk = —Ck — is the cardinality of the k-th cluster w.r.t. z1:n , and ?k is a distribution over component parameters that is independent from z. The variational construction in Eq.(10) and (11) is similar to Eq.(7) and Q (8), except for two significant differences: (1) p(z1:n —x1:n) is replaced by a product distribution i ?i (zi), and (2) ?—Ck , which depends on z1:n , is replaced by an independent distribution ?k . With this design, zi for different i and ?k for different k are independent w.r.t. q, thus resulting in a tractable predictive law below: Let q be a random probability measure given by Eq.(10) and (11), then K Pn X ? 0 0 i=1 ?i (k) ?(?) + ?k (?0). (12) Eq(D—?,?) [p(? —D)] = ?+n ?+n k=1

The approximate posterior has two sets of parameters: ? , (?1 , . . . , ?n) and ? , (?1 , . . . , ?n). With this approximation, the task of Bayesian learning reduces to the problem of finding the optimal setting of these parameters such that q(D—?, ?) best approximates the true posterior distribution. 4.3

Sequential Approximation

The first problem here is to determine the value of K. A straightforward approach is to fix K to a large number as in the truncated methods. This way, however, would incur substantial computational costs on unnecessary components. We take a different approach here. Rather than randomly initializing a fixed number of components, we begin with an empty model (i.e. K = 1)

and progressively refine the model as samples come in, adding new components on the fly when needed. Specifically, when the first sample x1 is observed, we introduce the first component and denote the posterior for this component by ?1 . As there is only one component at this point, we have z1 = 1, (1) i.e. ?1 (z1 = 1) = 1, and the posterior distribution over the component parameter is ?1 (d?) ? ?(d?)F (x1 —?). Samples are brought in sequentially. In particular, we compute ?i , and update ? (i?1) to ? i upon the arrival of the i-th sample xi . (i)

(i)

Suppose we have ? = (?1 , . . . , ?i) and ? (i) = (?1 , . . . , ?K) after processing i samples. To explain xi+1 , we can use either of the K existing components or introduce a new component ?k+1 . Then the posterior distribution of zi+1 , ?1 , . . . , ?K+1 given x1 , . . . , xn , xn+1 is p(zi+1 , ?1:K+1 —x1:i+1) ? p(zi+1 , ?1:K+1 —x1:i)p(xi+1 —zi+1 , ?1:K+1). (13) Using the tractable distribution q(?—?1:i , ? (i)) in Eq.(10) to approximate the posterior p(?—x1:i), we get p(zi+1 , ?1:K+1 —x1:i+1) ? q(zi+1 —?1:i , ? (i))p(xi+1 —zi+1 , ?1:K+1). (i+1)

(14)

Then, the optimal settings of qi+1 and ? that minimizes the Kullback-Leibler divergence between q(zi+1 , ?1:K+1 —q1:i+1 , ? (i+1)) and the approximate posterior in Eq.(14) are given as follows: ((i) R (i) wk ? F (xi+1 —?)?k (d?) (k ? K), R ?i+1 ? (15) ? ? F (xi+1 —?)?(d?) (k = K + 1), 4

Algorithm 1 Sequential Bayesian Learning of DPMM (for conjugate cases). Require: base measure params: ?, ?0 , observed samples: x1 , . . . , xn , and threshold Let K = 1, ?1 (1) = 1, w1 = ?1 , ?1 = ?(x1), and ?10 = 1. for i = 2 : n do Ti ? T (xi), and bi ? b(xi) (B(?k + Ti , ?k0 + ?) ? B(?k , ?k0) ? bi (k = 1, . . . , K) marginal log-likelihood: hi (k) ? B(? + Ti , ?0 + ?) ? B(?, ?0) ? bi (k = K + 1) P hi (k) hi (l) ?i (k) ? wk e / l wl e for k = 1, . . . , K + 1 with wK+1 = ? if ?i (K + 1) ¿ then wk ? wk + ?i (k), ?k ? ?k + ?i (k)Ti , and ?k0 ? ?k0 + ?i (k)? , for k = 1, . . . , K 0 wK+1 ? ?i (K + 1), ?K+1 ? ?i (K + 1)Ti , and ?K+1 ? ?i (K + 1)? K ?K +1 else P re-normalize ?i such that K k=1 ?i (k) = 1 wk ? wk + ?i (k), ?k ? ?k + ?i (k)Ti , and ?k0 ? ?k0 + ?i (k)? , for k = 1, . . . , K end if end for

(i)

with wk =

Pi

j=1

?j (k), and (i+1) ?k (d?)

(Qi+1 ?(d?) j=1 F (xj —?)?j (k) ? ?(d?)F (xi+1 —?)?i+1 (k)

(k ? K), (k = K + 1).

(16)

Discussion. There is a key distinction between this approximation scheme and conventional approaches: Instead of seeking the approximation of p(D—x1:n), which is very difficult (D is infinite) and unnecessary (only a finite number of components are useful), we try to approximate the posterior of a finite subset of latent variables that are truly relevant for prediction, namely z and ?1:K+1

. This sequential approximation scheme introduces a new component for each sample, resulting in n components over the entire dataset. This, however, is unnecessary. We find empirically that for most samples, ?i (K + 1) is negligible, indicating that the sample is adequately explained by existing component, and there is no need of new components. In practice, we set a small value and increase K only when ?i (K + 1) ¿ . This simple strategy is very effective in controlling the model size.

5

Algorithm and Implementation

This section discusses the implementation of the sequential Bayesian learning algorithm under two different circumstances: (1) ? and F are exponential family distributions that form a conjuate pair, and (2) ? is not a conjugate prior w.r.t. F . Conjugate Case.

In general, when ? is conjugate to F , they can be written as follows:

?(d?—?, ?0) = exp ?T ?(?) ? ?0 A(?) ? B(?, ?0) h(d?),

F (x—?) = exp ?(?)T T (x) ? ? A(?) ? b(x) .

(17) (18)

0

Here, the prior measure ? has a pair of natural parameters: (?, ?). Conditioned on a set of observations Pn x1 , . . . , xn , the posterior distribution remains in the same family as ? with parameters (? + i=1 T (xi), ?0 + n?). In addition, the marginal likelihood is given by Z F (x—?)?(d?—?, ?0) = exp (B(? + T (x), ?0 + ?) ? B(?, ?0) ? b(x)) . (19) ?

In such cases, both the base measure ? and the component-specific posterior measures ?k can be represented using the natural parameter pairs, which we denote by (?, ?0) and (?k , ?k0). With this notation, we derive a sequential learning algorithm for conjugate cases, as shown in Alg 1. Non-conjugate Case. In practical models, it is not uncommon that ? and F are not a conjugate pair. Unlike in the conjugate cases discussed above, there exist no formulas to update posterior 5

parameters or to compute marginal likelihood in general. Here, we propose to address this Qn issue using stochastic optimization. Consider a posterior distribution given by p(?—x1:n) ? ?(?) i=1 F (xi —?). A stochastic optimization method finds the MAP estimate of ? through update steps as below: ? ? ? + ?i (?? log ?(?) + n?? log F (xi —?)) .

(20)

The basic idea here is to use the gradient computed at a particular sample xi to approximate the true procedure converges to a (local) maximum, as long as the step size ?i satisfy P? gradient. ThisP ? 2 ? = ? and i=1 i i=1 ?i ¡ ?. Incorporating the stochastic optimization method into our algorithm, we obtain a variant of Alg 1. The general procedure is similar, except for the following changes: (1) It maintains point estimates of the component parameters instead of the posterior, which we denote by ??1 , . . . , ??K . (2) It computes the log-likelihood as hi (k) = log F (xi —??k). (3) The estimates of the component parameters are updated using the formula below: (i) (i?1) ??k ? ??k + ?i (?? log ?(?) + n?i (k)?? log F (xi —?)) .

Following the common practice of stochastic optimization, we set ?i = i ??
(21)
/n with ? ? (0.5, 1].

Prune and Merge. As opposed to random initialization, components created during this sequential construction are often truly needed, as the decisions of creating new components are based on knowledge accumulated from previous samples. However, it is still possible that some components introduced at early iterations would become less useful and that multiple components may be similar. We thus introduce a mechanism to remove undesirable components and merge similar ones. (i)

We identify opportunities to make such adjustments by looking at the weights. Let w ?k = Pi (i) P (i) (i) wk / l wl (with wk = ? (k)) be the relative weight of a component at the i-th iteraj=1 j tion. Once the relative weight of a component drops below a small threshold ?r , we remove it to save unnecessary computation on this component in the future. The similarity between two components ?k and ?k0 can be measured in terms of the distance bePi tween ?i (k) and ?i (k 0) over all processed samples, as d? (k, k 0) = i?1 j=1 —?j (k) ? ?j (k 0)—. We increment ?i (k) to ?i (k) + ?i (k 0) when ?k and ?0k are merged (i.e. d? (k, k 0) ¡ ?d). We also merge the associated sufficient statistics (for conjugate case) or take an weighted average of the parameters (for non-conjugate case). Generally, there is no need to perform such checks at every iteration. Since computing this distance between a pair of components takes O(n), we propose to examine similarities at an O(i ? K)-interval so that the amortized complexity is maintained at O(nK). Discussion. As compared to existing methods, the proposed method has several important advantages. First, it builds up the model on the fly, thus avoiding the need of randomly initializing a set of components as required by truncation-based methods. The model learned in this way can be readily extended (e.g. adding more components or adapting existing components) when new data is available. More importantly, the algorithm can learn the model in one pass, without the need of iterative updates over the data set. This distinguishes it from MCMC methods and conventional variational learning algorithms, making it a great fit for large scale problems.

6

Experiments

To test the proposed algorithm, we conducted experiments on both synthetic data and real world applications ? modeling image patches and document clustering. All algorithms are implemented using Julia [2], a new language for high performance technical computing. 6.1

Synthetic Data

First, we study the behavior of the proposed algorithm on synthetic data. Specifically, we constructed a data set comprised of 10000 samples in 9 Gaussian clusters of unit variance. The distances between these clusters were chosen such that there exists moderate overlap between neighboring clusters. The estimation of these Gaussian components are based on the DPMM below:

D ? DP ? ? N (0, ?p2 I) , ? i ? D, xi ? N (? i , ?x2 I). (22) 6

8
8
6
6
4
4
2
2
0
0
?2
?2
?4
?4
?8 ?8
?5.5 ?6
?6
?6
?4
?2
0
2
4
6
8
?8 ?8
?6
?4
?2
CGS
2
4
6
8
TVF
8
8
6
6
4
4
2
2
0
0
?2

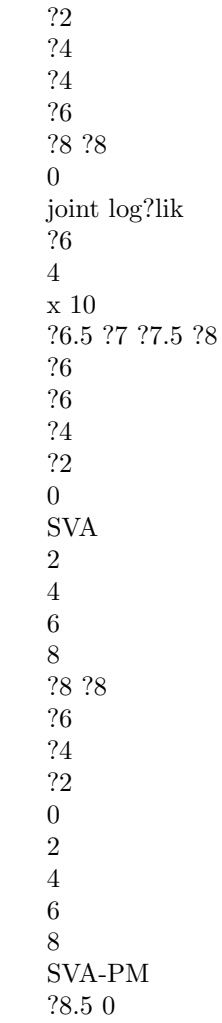

Figure 1: Gaussian clusters on synthetic data obtained using different methods. Both MC-SM and SVA-PM identified the 9 clusters correctly. The result of MC-SM is omitted here, as it looks the same as SVA-PM.

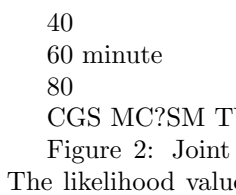

Figure 2: Joint log-likelihood on synthetic data as functions of run-time. The likelihood values were evaluated on a held-out testing set. (Best to view with color)

Here, we set ? = 1, ?p = 100 and ?x = 1. We tested the following inference algorithms: Collapsed Gibbs sampling (CGS) [12], MCMC with Split-Merge (MC-SM) [6], Truncation-Free Variational Inference (TFV) [21], Sequential Variational Approximation (SVA), and its variant Sequential Variational Approximation with Prune and Merge (SVA-PM). For CGS, MC-SM, and TFV, we run the updating procedures iteratively for one hour, while for SVA and SVA-PM, we run only one-pass. Figure 1 shows the resulting components. CGS and TFV yield obviously redundant components. This corroborates observations in previous work [9]. Such nuisances are significantly reduced in SVA, which only occasionally brings in redundant components. The key difference that leads to this improvement is that CGS and TFV rely on random initialization to bootstrap the algorithm, which would inevitably introduce similar components, while SVA leverages information gained from previous samples to decide whether new components are needed. Both MC-SM and SVA-PM produce desired mixtures, demonstrating the importance of an explicit mechanism to remove redundancy. Figure 2 plots the traces of joint log-likelihoods evaluated on a held-out set of samples. We can see that SVA-PM quickly reaches the optimal solution in a matter of seconds. SVA also gets to a reasonable solution within seconds, and then the progress slows down. Without the prune-and-merge steps, it takes much longer for redundant components to fade out. MC-SM eventually reaches the optimal solution after many iterations. Methods relying on local updates, including CGS and TFV, did not even come close to the optimal solution within one hour. These results clearly demonstrate that our progressive strategy, which gradually constructs the model through a series of informed decisions, is much more efficient than random initialization followed by iterative updating. 6.2

Modeling Image Patches

Image patches, which capture local characteristics of images, play a fundamental role in various computer vision tasks, such as image recovery and scene understanding. Many vision algorithms rely on a patch dictionary to work. It has been a common practice in computer vision to use parametric methods (e.g. K-means) to learn a dictionary of fixed size. This approach is inefficient when large datasets are used. It is also difficult to be extended when new data with a fixed K. To tackle this problem, we applied our method to learn a nonparametric dictionary from the SUN database [24], a large dataset comprised of over 130K images, which capture a broad variety of scenes. We divided all images into two disjoint sets: a training set with 120K images and a testing set with 10K. We extracted 2000 patches of size 32 ? 32 from each image, and characterize each patch by a 128-dimensional SIFT feature. In total, the training set contains 240M feature vectors. We respectively run TFV, SVA, and SVA-SM to learn a DPMM from the training set, based on the 7

550

?100 avg. pred. log?lik

avg. pred. log?lik

500 ?120 ?140 ?160 ?180

Figure 3: Examples of image patche clusters learned using SVA-PM. Each row corresponds to a cluster. We can see similar patches are in the same cluster.

0
2
4
6
TVF SVA SVA?PM 8
450 400 350 300 0
2
hour
Figure 4:
Average loglikelihood on image modeling as functions of run-time.
4
6 hour
8
TVF SVA SVA?PM 10
Figure 5:
Average loglikelihood of document clusters as functions of run-time.

formulation given in Eq.(22), and evaluate the average predictive log-likelihood over the testing set as the measure of performance. Figure 3 shows a small subset of patch clusters obtained using SVA-PM. Figure 4 compares the trajectories of the average log-likelihoods obtained using different algorithms. TFV takes multiple iterations to move from a random configuration to a sub-optimal one and get trapped in a local optima. SVA steadily improves the predictive performance as it sees more samples. We notice in our experiments that even without an explicit redundancy-removal mechanism, some unnecessary components can still get removed when their relative weights decreases and becomes negligible. SVM-PM accelerates this process by explicitly merging similar components. 6.3

Document Clustering

Next, we apply the proposed method to explore categories of documents. Unlike standard topic modeling task, this is a higher level application that builds on top of the topic representation. Specifically, we first obtain a collection of m topics from a subset of documents, and characterize all documents by topic proportions. We assume that the topic proportion vector is generated from a category-specific Dirichlet distribution, as follows D ? DP (? ? Dirsym (?p)) ,

? i ? D, xi ? Dir(?x ? i).

(23)

Here, the base measure is a symmetric Dirichlet distribution. To generate a document, we draw a mean probability vector ? i from D, and generates the topic proportion vector xi from Dir(?x ? i). The parameter ?x is a design parameter that controls how far xi may deviate from the categoryspecific center ? i . Note that this is not a conjugate model, and we use stochastic optimization instead of Bayesian updates in SVA (see section 5). We performed the experiments on the New York Times database, which contains about 1.8M articles from year 1987 to 2007. We pruned the vocabulary to 5000 words by removing stop words and those with low TF-IDF scores, and obtained 150 topics by running LDA [3] on a subset of 20K documents. Then, each document is represented by a 150-dimensional vector of topic proportions. We held out 10K documents for

testing and use the remaining to train the DPMM. We compared SVA, SVA-PM, and TVF. The traces of log-likelihood values are shown in Figure 5. We observe similar trends as above: SVA and SVA-PM attains better solution more quickly, while TVF is less efficient and is prune to being trapped in local maxima. Also, TVF tends to generate more components than necessary, while SVA-PM maintains a better performance using much less components.

7

Conclusion

We presented an online Bayesian learning algorithm to estimate DP mixture models. The proposed method does not require random initialization. Instead, it can reliably and efficiently learn a DPMM from scratch through sequential approximation in a single pass. The algorithm takes in data in a streaming fashion, and thus can be easily adapted to new data. Experiments on both synthetic data and real applications have demonstrated that our algorithm achieves remarkable speedup ? it can attain nearly optimal configuration within seconds or minutes, while mainstream methods may take hours or even longer. It is worth noting that the approximation is derived based on the predictive law of DPMM. It is an interesting future direction to investigate how it can be generalized to a broader family of BNP models, such as HDP, Pitman-Yor processes, and NRMIs [10]. 8

2 References

[1] C. Antoniak. Mixtures of dirichlet processes with applications to bayesian nonparametric problems. The Annals of Statistics, 2(6):1152?1174, 1974. [2] Jeff Bezanson, Stefan Karpinski, Viral B. Shah, and Alan Edelman. Julia: A fast dynamic language for technical computing. CoRR, abs/1209.5145, 2012. [3] David Blei, Ng Andrew, and Michael Jordan. Latent dirichlet allocation. Journal of Machine Learning Research, 3:993?1022, 2003. [4] David M. Blei and Michael I. Jordan. Variational methods for the Dirichlet process. In Proc. of ICML?04, 2004. [5] Michael Bryant and Erik Sudderth. Truly nonparametric online variational inference for hierarchical dirichlet processes. In Proc. of NIPS?12, 2012. [6] David B. Dahl. Sequentially-allocated merge-split sampler for conjugate and nonconjugate dirichlet process mixture models, 2005. [7] Nils Lid Hjort, Chris Holmes, Peter Muller, and Stephen G. Walker. Bayesian Nonparametrics: Principles and Practice. Cambridge University Press, 2010. [8] Matt Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic variational inference. arXiv eprints, 1206.7501, 2012. [9] Michael C. Hughes, Emily B. Fox, and Erik B. Sudderth. Effective split-merge monte carlo methods for nonparametric models of sequential data. 2012. [10] Lancelot F. James, Antonio Lijoi, and Igor Pr?unster. Posterior analysis for normalized random measures with independent increments. Scaninavian Journal of Stats, 36:76?97, 2009. [11] Kenichi Kurihara, Max Welling, and Yee Whye Teh. Collapsed variational dirichlet process mixture models. In Proc. of IJCAI?07, 2007. [12] Radford M. Neal. Markov Chain Sampling Methods for Dirichlet Process Mixture Mod-

els. Journal of computational and graphical statistics, 9(2):249?265, 2000. [13] David J. Nott, Xiaole Zhang, Christopher Yau, and Ajay Jasra. A sequential algorithm for fast fitting of dirichlet process mixture models. In Arxiv: 1301.2897, 2013. [14] Ian Porteous, Alex Ihler, Padhraic Smyth, and Max Welling. Gibbs Sampling for (Coupled) Infinite Mixture Models in the Stick-breaking Representation. In Proc. of UAI?06, 2006. [15] Carl Edward Rasmussen. The Infinite Gaussian Mixture Model. In Proc. of NIPS?00, 2000. [16] Jayaram Sethuraman. A constructive definition of dirichlet priors. Statistical Sinica, 4:639?650, 1994. [17] S.Jain and R.M. Neal. A split-merge markov chain monte carlo procedure for the dirichlet process mixture model. Journal of Computational and Graphical Statistics, 13(1):158?182, 2004. [18] Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical Dirichlet Processes. Journal of the American Statistical Association, 101(476):1566?1581, 2007. [19] Y.W. Teh, K. Kurihara, and Max Welling. Collapsed Variational Inference for HDP. In Proc. of NIPS?07, volume 20, 2007. [20] Chong Wang and David Blei. A split-merge mcmc algorithm for the hierarchical dirichlet process. arXiv eprints, 1201.1657, 2012. [21] Chong Wang and David Blei. Truncation-free stochastic variational inference for bayesian nonparametric models. In Proc. of NIPS?12, 2012. [22] Chong Wang and David M Blei. Variational Inference for the Nested Chinese Restaurant Process. In Proc. of NIPS?09, 2009. [23] Chong Wang, John Paisley, and David Blei. Online variational inference for the hierarchical dirichlet process. In AISTATS?11, 2011. [24] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In Proc. of CVPR?10, 2010.

9