Boosted Stochastic Backpropagation for Variational Inference

Ghassen Jerfel

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Master of Science in Engineering

Recommended for Acceptance
by the Department of
Computer Science
Adviser: Dr. Barbara Engelhardt

June 2017

Abstract

Variational inference has risen in popularity with the advent of deep generative models due to its efficient and scalable approximation of the posterior distribution. However, VI is not generally guaranteed to capture the true posterior.

In this paper, we propose a mixture-based non-parametric variational inference algorithm. We prove a convergence to the true posterior in $O(1/t)$ where t is the number of mixture components.

Using a mixture of Gaussians as the variational approximation, we propose boosted stochastic backpropagation where we derive tractable approximations and practical insights to avoid numerical instability when learning a new component in the mini-batch setting.

We then use boosted stochastic backpropagation as an unsupervised boosting meta-algorithm for non-parametric density estimation and apply it to Variational Autoencoders.

We empirically demonstrate the advantage of flexible and multimodal posterior approximations in density estimation on MNIST.

Acknowledgements

I would like to thank Dr. Barbara Engelhardt for her guidance throughout the past 2.5 years of academic research. I would also like to thank Dr. Mehmet Basbug for his mentorship and help both academically and personally even after his graduation.

I would like to thank the Compute Science department for allowing me to spend one more year at this great institution with the company of stellar peers and senior researchers.

Finally, I'd like to thank my family and friends for their support throughout the past five years.

To Baba, Mama, Rima, and Didou.

Contents

Abstract iii

Acknowledgements iv

List of Figures ix

1 Introduction **1**

2 Related Work **4**

2.1 Advances In Variational Inference 4

2.2 Mixture Models in Variational Inference 5

2.3 Boosting in Variational Inference and Deep Generative Models 5

3 Stochastic Backpropagation for Variational Inference **7**

3.1 Variational Inference 7

3.2 Stochastic Backpropagation 8

3.3 Choice of Variational Approximations 8

4 Boosted Stochastic Backpropagation **11**

4.1 Mixture Variational Approximations 11

4.2 Greedy Mixture Building 12

4.3 Boosted Gaussian Backpropagation 14

4.4 Theoretical Guarantees 15

4.5 Stopping Condition 16

5 Boosting Variational Autoencoders **18**

5.1 Boosting Deep Latent Gaussian Models 18

5.2 Architecture 19

5.3 Numerical Stability 20

6 Experiments **22**

6.1 Experiment Design 22

6.2 Results 23

7 Conclusion **26**

7.1 Summary 26

7.2 Future Work 27

A Computational Complexity **28**

B Evidence Lower Bound **29**

C Objective Derivation and Theoretical Guarantees **31**

C.1 The Functional Gradient 31

C.2 Alpha Gradients 32

C.3 Gradient Boosting Objective 33

C.4 Joint Convexity of the KL divergence 33

C.5 Strong Smoothness of the KL divergence 35

D Analytical Mixture Bounds and Practical Tricks **36**

D.1 Mixture Cross-Entropy Lower Bound 36

D.2 Mixture Cross-Entropy Upper Bound 37

D.3 Mixture Likelihood and LogSumExp 38

E Useful Gaussian Identities **39**

E.1 L_2 Norm Identity for Gaussian Distributions 39

E.2 Gaussian Cross-Entropy 40
E.3 Analytical KL 40
E.4 Gaussian Mixture Statistics 41

Bibliography **42**

List of Figures

3.1 Deep Latent Gaussian Models [32] 9
3.2 Illustration of the deterministic Reparametrization (by Jaan Altosaar) 10

5.1 Deep generative models (e.g. GMVAE) capture a richer latent representation than other generative models such as Gaussian Mixture Mixtures 19
5.2 A mixture of Gaussians VAE model 19

6.1 Comparison of the complementary reconstructions 23
6.2 Comparison of the ELBO for baseline models 24
6.3 t-SNE visualization of the latent space: a more disentangled representation for the mixtur 24

Chapter 1

Introduction

Deep generative models, such as variational autoencoders (VAE) [20, 32], have risen in popularity due to their tractable likelihood lower-bound, ease of sampling, and state-of-the-art performance on natural image datasets [19]. VAEs couple the regularized latent variable representations with nonlinear likelihoods to learn flexible representations of complex high-dimensional data. Variational inference (VI) is then used as a probabilistic framework to shape a principled cost function and an intrinsic architecture for VAEs. [3] VI can scale to larger datasets more efficiently than other probabilistic inference algorithms such as Markov Chain Monte Carlo since it treats posterior inference as an optimization problem over a parametric family of approximate posterior distributions. [3]

In deep generative models, we can efficiently compute unbiased and low-variance Monte Carlo estimates of the gradient of the optimization objective without a variational EM algorithm. [3] This is possible due to the reparametrization trick. [20] The resulting algorithm is stochastic backpropagation [32] or stochastic gradient variational Bayes (SGVB)[20] which is the current best practice in VI due to its computational efficiency.

Most current implementations of variational inference are not generally guaranteed to retrieve the true posterior regardless of the runtime, unlike MCMC. [3] In most standard applications, variational approximations are chosen from a family of factorizable distributions such as diagonal Gaussians. These approximations fail to recover the posterior covariance and thus underestimate the posterior variance [3] in addition to failing to naturally capture posterior multimodality. These disadvantages are tolerated in practice due to the tractability of such approximations. However, it is unclear how to perform a finer trade-off between computational performance and statistical accuracy in VI which is common for other algorithms such as MCMC.

Nonlinear transformations in deep generative models can improve the flexibility of unimodal variational approximation in order to better capture the latent representation. Nonetheless, they are not theoretically guaranteed to capture the true posterior. In practice, the use of unimodal distributions, such as diagonal Gaussians in stochastic backpropagation, has been shown to hinder density estimation as witnessed in the over-pruning problem of VAEs. [35] Additionally, nonlinearities can capture some multimodality in the latent space by marginalizing the distribution of the top hidden layer. However, this latent representation suffers from local smoothing [34] and mode symmetry which lead to difficulties in identifying which modes correspond to which latent features [28]. This limits the performance on discriminative tasks such as clustering and classification [8].

Designing richer and more flexible posterior approximations for variational inference and deep generative models has been an active area of research. However, most approaches either lack the statistical consistency guarantees or don't allow for a flexible nonparametric density estimation as explained in the related works section (2). Mixture approximations, on the other hand, are intuitive to implement in a non-parametric way, and can model both arbitrarily smooth distributions and strong

modality. This approximation can thus improve both the generative and discriminative performance of unsupervised models.

In this paper, we propose a non-parametric variational approximation with Gaussian mixtures which provides an intuitive trade-off between the computational tractability and density estimation accuracy. Based on Breimans [5] observation that boosting can be viewed as a greedy optimization of a convex loss function, we develop boosted stochastic backpropagation as an unsupervised boosting meta-algorithm for deep latent Gaussian models. This combines the flexibility of nonlinear transformations with the incremental refinement and multimodality of non-parametric mixtures to efficiently retrieve rich posterior approximations.

In section 3, we introduce stochastic backpropagation as the best current practice in variational inference.

In section 4 we propose Gaussian mixtures as a variational approximation. We leverage T. Zhengs [37] approximation framework to prove the retrievability of any smooth posterior distribution at a rate of $O(1/t)$ as a function of the number of mixture components. We then derive analytical bounds and efficient identities for boosted stochastic backpropagation. We also propose a novel optimization method to the optimization of the component and its weight on a per-minibatch basis.

In section 5 we apply boosted stochastic backpropagation to deep latent Gaussian models in order to optimally add weighed encoder-decoder pairs into a mixture of VAEs and improve the overall KL divergence. We also address the numerical instability issues of mixtures and boosting which is caused by high-dimensionality and subsampling noise.

We then demonstrate, in section 6 the practical advantage of more flexible and multimodal posteriors on density estimation for MNIST.

Chapter 2

Related Work

The bulk of recent variational inference work has focused on designing more flexible posterior approximations. This line of research extends to the recent development of deep generative models.

2.1 Advances In Variational Inference

Rezende and Mohamed [31] propose normalizing flows (NF) which map a simple unimodal distribution through a sequence of invertible nonlinear transformation. Inverse auto-regressive flows [19] are an example of normalizing flows which achieves state-of-the-art performance on several density estimation tasks. However, these flows tend to be difficult to design and implement which might explain the very limited number of known transformations.

Combining Monte Carlo Markov Chain sampling with VI [33, 6] has displayed competitive performance but tends to be computationally expensive and difficult to scale to real-world problems.

Auxiliary variable variational models [25, 30] were used to enrich the variational approximation and better estimate the posterior covariance. However, it is unclear how the number of posterior modes relates to the design of such methods.

2.2 Mixture Models in Variational Inference

Mixture models were considered for flexible posterior approximation in early VI literature [12, 17, 2, 38]. These works established some theoretical guarantees for the monotonic improvement of the KL divergence with each additional mixture component. Mixture approximations were recently explored for deep generative models [8, 29, 34]. However, these works did not demonstrate an improvement in the log likelihood due to the non-informative prior. Overall, mixtures with a fixed number of components are sensitive to the initialization and require the evaluation of the likelihood and gradients of all components for each parameter update.

The a priori fixed number of mixture components is also a major disadvantage of these approaches since we might need an arbitrarily large number of components to approximate the true posterior within some inaccuracy tolerance ϵ. Therefore, a non-parametric density estimation and mixture building approach is more suitable for asymptotic guarantees and ease of tuning or early termination.

2.3 Boosting in Variational Inference and Deep Generative Models

Boosting variational inference was recently explored by Miller et al. [27] and Guo et al. [15]. However, [27] makes strong assumptions when deriving the gradient of the ELBO for each new component thus compromising the convergence of the algorithm. Additionally, the mixture weights are learned with a computationally expensive EM algorithm for each boosting iteration. [15] is motivated by the same greedy approximation framework by T. Zheng [37] and thus proves similar properties of the KL loss function. However, [15] deviates from the general gradient boosting framework and proposes a heuristic, the Laplacian Gradient Boosting, which requires computation-

ally expensive finite-differences estimates of the Hessian. Additionally, [15] does not consider stochastic backpropagation which requires the reparametrization of the gradient boosting objective. Overall, both works focus on approximating a known target distribution rather than learning the hidden representation of generative models. Accordingly, these works do not naturally extend to deep generative models where we jointly learn the generative and inference parameters with minibatches. Furthermore, both works were not applied to real-world datasets or benchmarks such as MNIST.

Grover and Ermon [14] propose a multiplicative boosting approach for generative models that does not enjoy the same theoretic guarantees as additive models, suffers from intractable likelihoods, and is difficult to sample from. Tolistikhin et al. [36] propose a provably optimal reweighing scheme for building a mixture of GANs. However, [36] requires heuristical approximations of the mixture coefficients. Additionally, [36] does not naturally extend to unsupervised models which lack the GAN discriminator's binary label accuracy output.

Chapter 3

Stochastic Backpropagation for Variational Inference

3.1 Variational Inference

Let x be the observed variables, z the latent variables and $p_\theta(x, z)$ the parametric model of their joint distribution, called the generative model. Performing inference requires marginalizing over the latent variables to compute the likelihood of the probabilistic model. However, the evaluation and differentiation of the marginal likelihood integral in (3.1) is usually intractable.

$$\log p_\theta(x) = \log \int_z p_\theta(x|z) p_\theta(z) dz \tag{3.1}$$

Variational inference introduces an approximate posterior $q_\phi(z|x)$ from a parametric family of distributions and provides a lower-bound on the marginal likelihood 3.2, the so-called Evidence Lower BOund (ELBO). [3] VI then formulates the inference problem as a minimization of the variational lower bound which allows the joint optimization of the generative parameters θ and the variational parameters ϕ with

EM-like updates. [3]

$$\log p_\theta(x) \geq \mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(x, z) - \log q_\phi(z|x)\right] \tag{3.2}$$

As shown in (B.2) in the appendix, minimizing this objective is equivalent to minimizing a Kullback-Leibler (KL) [22] divergence between the true posterior $p(z|x)$ and the approximate posterior $q_\phi(z|x)$. [3]

3.2 Stochastic Backpropagation

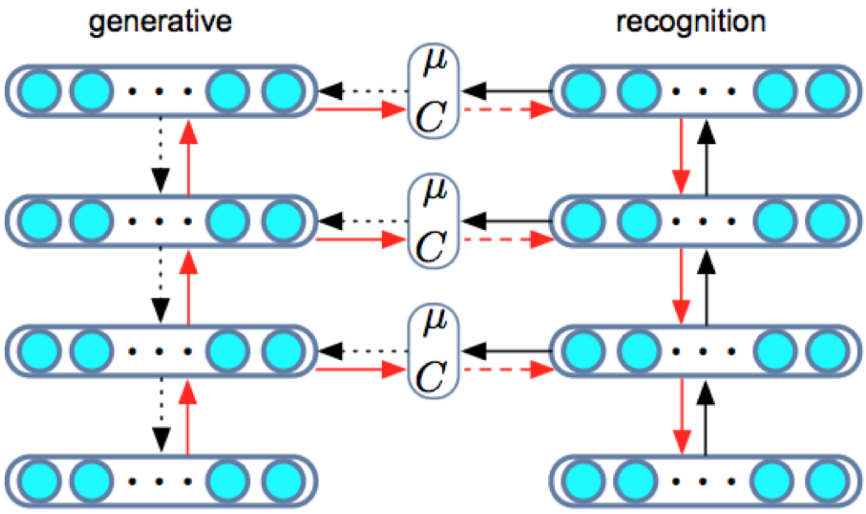
For continuous probabilistic models, the optimization of the ELBO can be done without resorting to the classic variational EM algorithm by using a deterministic reparametrization of the expectation (3.2) with respect to the variational parameters (3.3). This provides unbiased and empirically low-variance Monte Carlo estimates of the gradients [10] which can be used with general stochastic optimization methods such as SGD.

The resulting algorithm is known as stochastic backpropagation [32] which has become the best practice in variational inference due to its scalability and suitability for general inference in deep generative models (fig. 3.2). [31]

3.3 Choice of Variational Approximations

The KL divergence (eq. B.1) is strictly positive for $P \neq Q$ and equal to 0 only for $Q = P$. As a result, finding the general Q which minimizes the KL divergence is no easier than the original inference task, which we assume is intractable. The usual strategy therefore is to place simplifying constraints on Q, the most popular, due to its simplicity, being the mean field approximation. [31]

Figure 3.1: Deep Latent Gaussian Models [32]



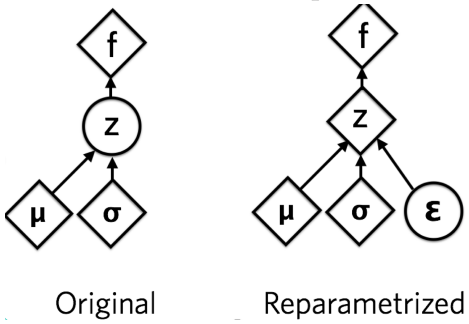
In the case of stochastic backpropagation, the reparametrization requirement limits the choice of the approximation to handful of distributions. [20] In practice, stochastic backpropagation is usually derived under the Gaussian approximation which guarantees an analytical form of the KL term (eq. E.4). Additionally, the log-likelihood in (eq. 3.2) can be rewritten using the location-scale transformation (fig. 3.3) for the Gaussian distribution as:

$$\mathbb{E}_{q_\phi(z|x)}[\log\left(p_\theta(x|z)\right)] = \mathbb{E}_{\mathcal{N}(\epsilon|0,I)}[\log\left(p_\theta(x|\mu_\phi(x) + \sigma_\phi \odot \epsilon)\right)] \tag{3.3}$$

This expectation cannot be solved analytically for most expressive models. Therefore, we compute the gradients with respect to the variational parameters ϕ with Monte Carlo estimates:

$$\nabla_{\theta,\phi}\mathbb{E}_{q_\phi(z|x)}[\log\left(p_\theta(x|z)\right)] = \mathbb{E}_{\mathcal{N}(\epsilon|0,I)}[\nabla_{\theta,\phi}\log\left(p_\theta(x|\mu_\phi(x) + \sigma_\phi \odot \epsilon)\right)] \tag{3.4}$$

Figure 3.2: Illustration of the deterministic Reparametrization (by Jaan Altosaar)



Chapter 4

Boosted Stochastic Backpropagation

4.1 Mixture Variational Approximations

While there has been research on enriching unimodal distributions to better capture high-dimensional latent structures [20, 6], the resulting algorithms are not generally guaranteed to retrieve the true posterior. Additionally, it is usually unclear how to control the modality of the representation or how to map the resulting modes to their corresponding latent features which hinders the discriminative quality of the model. [28]

On the other hand, mixture models (4.1) are easy to design, are inherently multimodal and can capture several properties such as heteroscedasticity even when the base distributions are homoscedastic. [38]

Therefore, to model complex distributions, we can use the t-component variational mixture:

$$co_t(\mathbb{F}) = F = \sum_{i=1}^{t} \alpha_i F_i \ , \sum \alpha_i = 1 \ , \alpha_i \geq 0 \tag{4.1}$$

For base distributions in $\mathbb{F}$, the set of all finite mixture models defines the convex hull of $\mathbb{F}$:

$$conv(\mathbb{F}) = \bigcup_{k=1}^{\infty} \left\{ \sum_{i=1}^{k} \alpha_i F_i : \alpha_i \geq 0, \sum_{i=1}^{k} \alpha_i = 1, F_i \in \mathbb{F} \right\} \tag{4.2}$$

For the remainder of this paper, we will focus on finite Gaussian mixtures due to the ubiquity of Gaussian assumptions, their easy reparametrization and their analytical KL divergence. Additionally, the convex hull of Gaussian distributions can include any arbitrarily smooth distribution. [26] Therefore, Gaussian mixtures can capture both heavy tails and strong modality, asymptotically in the number of mixture components. Even more interestingly, a mixture of restricted Gaussian such as locked-covariance or diagonal Gaussians, which are common for deep generative models, can still retrieve the true distribution but might require more components. [38]

Note that it should be trivial to extend this framework to other reparametrizable distributions for use with stochastic backpropagation.

4.2 Greedy Mixture Building

At each step t, we select a new distribution from the base family $\mathbb{F}$ that we admixture into the existing convex combination with a coefficient α while re-scaling the existing mixture by $1 - \alpha$. The resulting mixture (4.3) is a convex hull of $\mathbb{F}$: $co_t(\mathbb{F})$ spanned by t components:

$$F_t = (1 - \alpha)F_{t-1} + \alpha Q_t \tag{4.3}$$

The new component and its weight are selected to minimize the KL divergence of the resulting convex combination:

$$Q_t^*, \alpha^* = \arg\min_{Q_t, \alpha} KL((1 - \alpha)F_{t-1} + \alpha Q_t \| P) \tag{4.4}$$

This iterative process allows a fine trade-off between the runtime and the accuracy as the algorithm can theoretically search all of $conv(\mathbb{F})$ (4.2) and terminate when the overall accuracy reaches some predefined ϵ. This non-parametric approach is thus guaranteed to retrieve the true posterior or an ϵ-approximate posterior as long as the posterior is in $conv(\mathbb{F})$ which we assume is valid for $\mathbb{F}$ consisting of Gaussian distributions. [26]

This is a greedy stage-wise strategy for mixture building that is somewhat equivalent to boosting as first noted by Breiman [5] and later formalized by Friedman [9] in the Gradient Boosting framework.

In [9], boosting is interpreted as functional gradient descent on a loss function where the functional gradient is defined as the direction of maximal change in the loss function when adding an infinitesimal perturbation to the input.
A functional gradient $\nabla E[f]$ is thus implicitly defined as the linear term of the change in a function due to a small perturbation ϵ: $E[f + \epsilon g] = E[f] + \epsilon \langle \nabla E[f], g \rangle + O(\epsilon^2)$.

For the KL loss function, we derive the functional gradient in the appendix (eq. C.1) resulting in:

$$\nabla_{KL(F_{t-1}\|P)} = \log f_{t-1} - \log p \tag{4.5}$$

The negative gradient $-\nabla_{KL(F_{t-1}\|P)}$ is said to define the steepest descent in the functional space. [9] Therefore, at each iteration, we perform a step of restricted gradient descent, within the class of base distributions, since we cannot follow the gradient directly and instead replace it with a search direction from a set of allowable descent directions. The step size is then determined using a line search (4.6) along the steepest direction to find the optimal coefficient for the newly selected distribution before adding it to the current mixture.

$$\alpha^* = \arg\min_{\alpha} KL((1-\alpha)F_{t-1} + \alpha Q_t \| P) \tag{4.6}$$

Notice that the gradient projection onto the base distribution set could degenerate to a point mass that maximizes $\int_x q \log \frac{f_{t-1}}{p}$. Therefore, we follow the well-established regularization scheme in [9] to select each new component which results in the following optimization problem:

$$\arg\min_{Q_t} KL(Q_t\|P) - KL(Q_t\|F_{t-1}) + \frac{\lambda}{2}\left\|q_t^2 - q_t f_{t-1}\right\|^2 = \arg\min_{Q_t} KL(Q_t\|P) - KL(Q_t\|F_{t-1}) + \frac{\lambda}{2}\int_x q_t^2 - q_t f_{t-1} \tag{4.7}$$

Optimizing (4.7) could be interpreted as biasing the learning of a new component to be different from the previous mixture, similarly to the idea of reweighing the negative samples in AdaBoost. [9] In this case, the L_2 term reconciles the two different KL terms.

4.3 Boosted Gaussian Backpropagation

If we limit our base distributions to the Gaussian family, we can simplify the objective (4.7) by expressing the squared L_2 distance analytically (as derived in the appendix eq. E.2):

$$\int_x q_t^2 - q_t f_{t-1} = \int \mathcal{N}(\mu_t; \mu_t, 2 * \Sigma_t) - \sum_{i=1}^{t-1} \alpha_i \int \mathcal{N}(\mu_t; \mu_i, \Sigma_t + \Sigma_i) \tag{4.8}$$

Furthermore, unlike typical Gaussian mixture models, we sidestep the non-differentiability of the discrete sampling step by marginalizing over the different mixture components which perfectly suits our weighed summation model. Accordingly, we can re-write the expectation under the mixture as a weighed sum of expectations (eq. 4.9) under unimodal Gaussian components, thus resulting in a

variant of stochastic backpropagation:

$$\mathbb{E}_{f_t}[\log p(x|z) - \log p(z) - \log f_t(z|x)] = \sum_{i=1}^{t} \alpha_i \mathbb{E}_{f_i}[\log p(x|z) - \log p(z) - \log f_t(z|x)] \tag{4.9}$$

Another useful consequence of our two-step steepest descent optimization is the convexity of the linear in (4.6) due to the non-negative second derivatives in (4.10). Therefore, we can tractably estimate the gradients using Monte Carlo estimates under the expectation of individual components (as derived in the appendix eq. C.2):

$$\begin{aligned}
\nabla_\alpha KL((1-\alpha)F_{t-1} + \alpha Q_t \| P) &= \mathbb{E}_{q_t}[\log \frac{(1-\alpha)f_{t-1} + \alpha q_t}{p}] \\
&\quad - \sum_{i}^{t} \alpha_i \mathbb{E}_{f_i}[\log \frac{(1-\alpha)f_{t-1} + \alpha q_t}{p}] \\
\nabla_\alpha^2 KL((1-\alpha)F_{t-1} + \alpha Q_t \| P) &= \int_x \frac{[f_{t-1} - q_t]^2}{(1-\alpha)q_t + \alpha f_{t-1}}
\end{aligned} \tag{4.10}$$

Notice that, in the stochastic setting, we alternate between gradient descent on (4.6) and on (4.7) per minibatch as this has demonstrated better numeric stability than the naive implementation and is quite intuitive for dealing with outliers and degenerate components that the L_2 regularization term might not mitigate (which is the case for any dimensionality $d > 2$).

4.4 Theoretical Guarantees

For a convex and strongly smooth loss function, T. Zheng's [37] greedy approximation framework provides a theoretical guarantee for the convergence to a target distribution that's in the convex hull of the base family at a rate of $O(1/t)$. [37] also allows for non-optimal components at each step provided that the discrepancy between the selected component and the optimal solution tends to 0 with the number of iterations.

This framework perfectly suits our setup where, at each step, we might not be able to find the exact optimum since the objective in (C.15) is not necessarily convex. However, the KL loss function is jointly-convex in its inputs, as proven in the appendix (eq. C.11). We also prove that the KL divergence is strongly smooth under the assumption of lower-bounded densities in the appendix (eq. C.11).

Note that Li et. al [24] demonstrated an optimal convergence rate of $O(1/t)$ for the iterative process of mixture density estimation when guided by the minimization of the maximum likelihood or the KL divergence. However, in the case of the KL divergence, [24] assumes the boundedness of the log ratio of densities in $\mathbb{F}$ instead of the boundedness of the densities themselves. This might not be practical since two Gaussian densities with the same variance and different means results in a log density ratio that tends to infinity at the tails on a non-compact support. Therefore, our boundedness assumption, which requires a compact support domain, for the base distributions is practically justified.

4.5 Stopping Condition

If at an iteration t the current F_{t-1} cannot be improved by adding any of the base densities, the algorithm terminates.

To better understand this termination condition, let's derive the functional gradient in terms of the α gradients (C.2) as α tends to 0 since this corresponds to the maximum possible perturbation we can introduce to the loss function of the current mixture:

$$\begin{aligned}\alpha \searrow 0 \rightarrow \nabla_\alpha KL((1-\alpha)F_{t-1} + \alpha Q_t \| P) &= \int_x q_t \log \frac{f_{t-1}}{p} - \int_x f_{t-1} \log \frac{f_{t-1}}{p} \\ &= KL(F_{t-1}\|p) - KL(Q_t\|P) - KL(Q_t\|F_{t-1})\end{aligned} \tag{4.11}$$

It is easy to check that the expression in eq. 4.11 is equivalent to that of the negative gradient projection which we seek to maximize in (4.7) minus the L_2 regularization. The improvement of the KL divergence with respect to the admixture of a new component at an iteration t is thus guaranteed if the expression (4.11) is negative.

Since the second derivative (4.10) is non-negative for all values of α: if $KL(F_{t-1}\|P) \geq KL(Q_t\|P)$ and (4.11) is positive, all mixtures $(1-\alpha)F_{t-1} + \alpha Q_t$ would have a higher KL divergence than the previous mixture independently of the optimality of F_{t-1}.

This inequality could also be considered as a characterization of the expressivity of the base distributions family $\mathbb{F}$ such that, to guarantee convergence, a base family would need to satisfy:

$$KL(Q_t\|P) - KL(F_{t-1}\|P) \leq KL(Q_t\|F_{t-1})$$

for all $F_{t-1} \in conv(\mathbb{F})$ and $Q_t \in \mathbb{F}$ (assuming P is also $\in conv(\mathbb{F})$).

Chapter 5

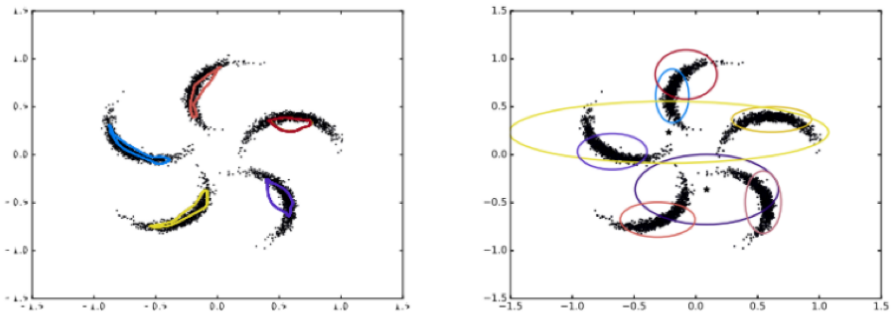
Boosting Variational Autoencoders

5.1 Boosting Deep Latent Gaussian Models

As discussed earlier, a nonlinear transformation can enrich a simple distribution (fig. 5.1) and allow for a more accurate representation of the latent space. Furthermore, the use of inference networks to "amortize" variational inference is an important practice that leverages the computational efficiency of back-propagating through neural networks to learn inverse mapping from observations to the latent space. [11] This accelerates the training and testing times as well as better scales the inference to larger datasets. [20]

Therefore, the best current practice for variational inference is a combination of fast amortized inference with stochastic backpropagation which is most suitable for deep latent Gaussian models such as VAEs. Note that in most VAE implementations, the prior $p_\theta(z)$ is a spherical Gaussian $\mathcal{N}(z|0, I)$ and the variational approximation is a diagonal Gaussian distribution $q_\phi(z|x)$. This unimodal and factorizable parametrization has been shown to cause over-pruning in the latent dimensions which hinders density estimation. [35, 7]

Figure 5.1: Deep generative models (e.g. GMVAE) capture a richer latent representation than other generative models such as Gaussian Mixture Mixtures

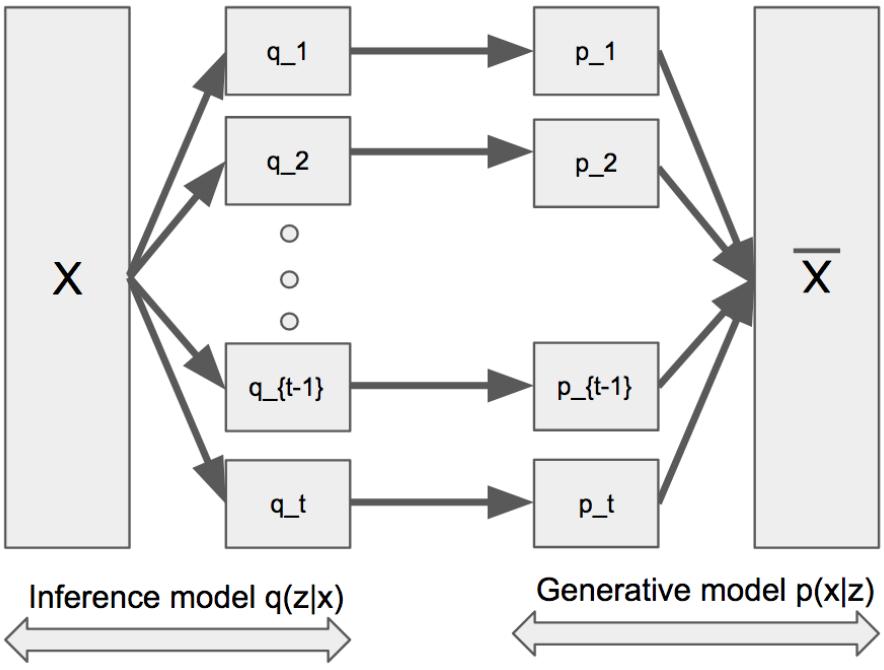


(b) Density of GMVAE (c) Density of GMM

Therefore, VAEs are the perfect application for boosted stochastic backpropagation using a mixture of diagonal Gaussian distributions which can be combined with VI advances such as normalizing flows [31] or importance weighed sampling [6].

5.2 Architecture

Figure 5.2: A mixture of Gaussians VAE model



Boosted stochastic backpropagation is used as a meta-algorithm for the selection and admixture of baseline encoder-decoder pairs into a mixture of VAEs (fig. 5.2) while holding the previously-trained autoencoders fixed. We train a new decoder for each encoder due to the coupling of their optimization and the known issues of forgettability [21] in neural networks. Accordingly, if we were to keep the same decoder throughout progress of our algorithm, the decoder would not be able to generate relevant samples from the latent representation of earlier encoders.

We perform sampling in a reparametrizable fashion by generating a single sample from each component or autoencoder and then taking a weighted average. This can preserve the covariance among the latent dimensions and between the different components since we have to evaluate an approximation of the pair-wise entropy for each new encoder. If, however, we sought a computational compromise, we could represent the mixture of the latent samples as a Gaussian sampling with matching first and second moments which were computed in (E.4).

A discussion of the computational complexity is provided in the appendix (A).

5.3 Numerical Stability

In the general setting of VAEs, the neural network parametrization makes the boosting optimization a bit more difficult since each data point maps to a different Gaussian representation. This induces some noise into our residual fitting algorithm. Additionally, for dimensions $d >$, the L_2 regularization term vanishes since it's not in the log domain. Furthermore, due to minibatching with equal weights (since we should not require reweighing when we are already performing gradient boosting), our early experiments suffered from numerical instability and divergence problems as each new component was extremely sensitive to the uniformly sampled minibatch which could be well covered by the previous component whereas the next minibatch could a com-

plete outlier to the rest of the dataset. These problems are not unique to our optimization formulation as they seem to be recurrent in Mixture Density Networks as well.[4]

To mitigate such problems, we first used Batch Normalization [16] to re-scale the contribution of each batch and generally accelerate the VAE training as proposed in [35].

However, our most successful practical contribution has been to alternate the inference of the new component (thus the steepest descent) and the mixture weight (the line search) on a per-minibatch basis rather than fully training a new component and then proceeding to fitting a mixture weight to that component. This modification should not invalidate the convergence guarantees of the framework in [37] but warrant further theoretical analysis that is outside the scope of the current version of this paper.

Chapter 6

Experiments

6.1 Experiment Design

For our experiments, we parametrized our VAEs with two nonlinear (with softplus activation) hidden layer for each of the encoder and the decoder. We then connect these layers to a 2 linear layers: one for the log of the variance and another for the mean of the Gaussian parametrization. These layers were taken at a size of 500 units as is common in the VAE literature [35].

All parameters were initialized using the Glorot and Bengio scheme [13].

For training, we used the ADAM [18] optimization algorithm with a learning rate of 0.001 and a batch size of 100. The models were implemented in Python using TensorFlow [1] and the code should be available soon.

Our experiments in this version are limited to MNIST, the standard handwritten digits dataset composed of 28x28 grayscale images (mapped to a 0-1 range) and consisting of $55,000$ training samples, $5,000$ validation samples to terminate the boosting algorithm in case of overfitting, and $10,000$ testing samples [23].

We cite our numerical instability issues as the main cause of our lack of substantial experimentation as we did not formalize the process described in (5.3) until a few

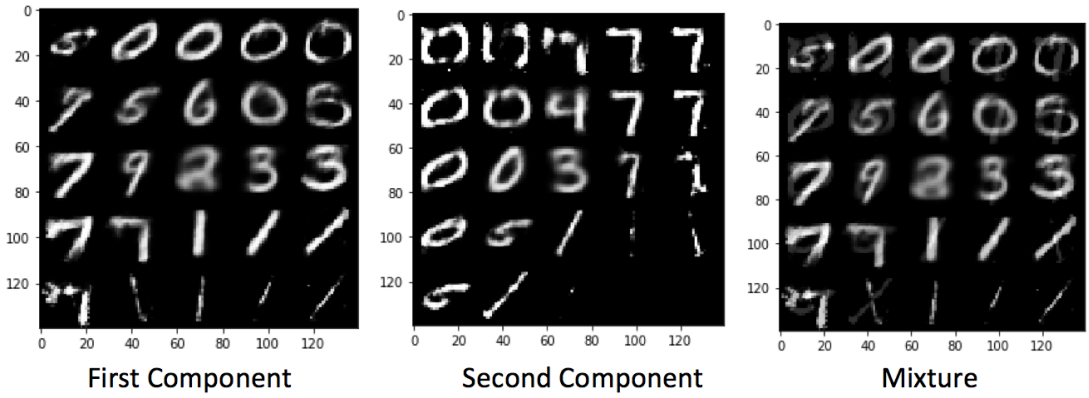
hours before the submission deadline. However, it is worthy to note that none of the existing regularization schemes for gradient boosting scaled to the MNIST dimensionality which might explain the limitation of existing literature to 2-dimensional toy datasets and shallow models.

6.2 Results

We first evaluate the log-likelihood using importance sampling as described in [6] with 1000 Monte Carlo samples at each iteration of our boosting algorithm and noticed a monotonic improvement in the test log likelihood on baseline VAEs. A small example of such improvement can be seen in the convergence values for the models in fig. 6.2.

We noticed an improvement of the reconstruction error as show in fig. (6.2) and a somewhat more disentangled latent representation as show in the t-SNE visualization (fig. 6.2). Due to the marginal and almost unnoticeable improvement in generated samples, we do not report qualitative results.

Figure 6.1: Comparison of the complementary reconstructions



Early experiments that we do not report due to the recent change in our framework combined gradient boosting with reweighing process similar to that suggested in [15] and demonstrated the adversary effect of combining reweighting and gradient boosting. Early results also demonstrated a much better improvement, later in the

Figure 6.2: Comparison of the ELBO for baseline models

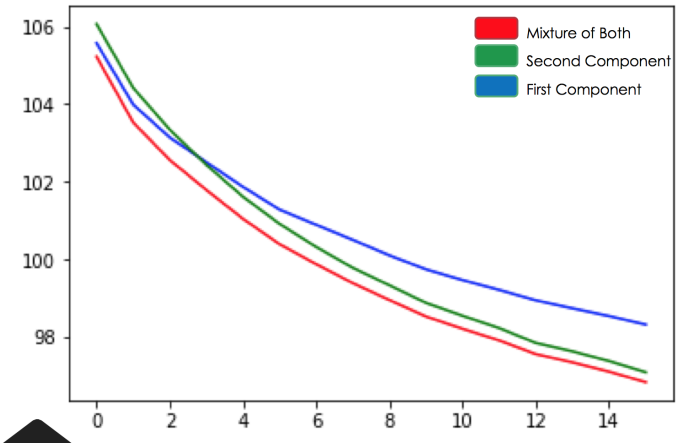
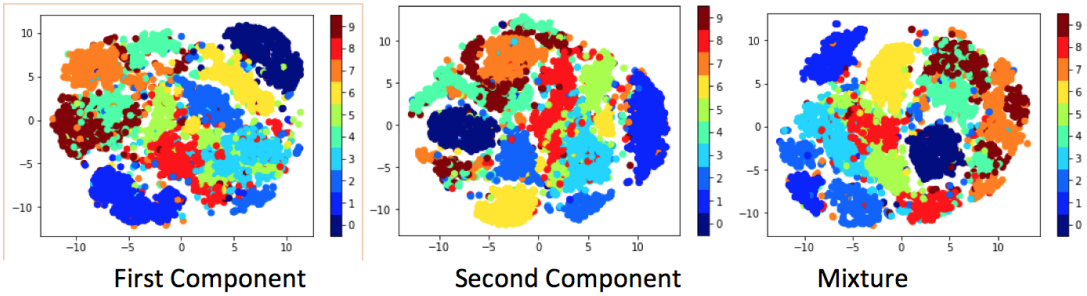


Figure 6.3: t-SNE visualization of the latent space: a more disentangled representation for the mixtur



training process of VAEs, when adding mixture components as compared to running the inference algorithm for a thousand or more extra epochs. This is motivated by the theoretical guarantees established in [2] of the monotonic improvement for each added mixture component when the neural network parametrization of the diagonal Gaussian cannot generalize any further. A similar observation was made with regards to adding depth or width to the neural networks, similarly to the observation made in [14].

Overall, our framework offers the theoretical guarantees of statistical consistency and inherent multimodality. In the case of MNIST, after 7 iterations of adding new components and rescaling older mixtures, we retrieved 7 components that seem to

specialize in specific digits. This can clearly enhance the performance on discriminative tasks.

Chapter 7

Conclusion

7.1 Summary

In this paper, we propose a both practical and theoretically founded unsupervised boosting algorithm variational inference that is guaranteed to converge with a rate of $O(1/t)$ to arbitrarily smooth posteriors.

Using a mixture of Gaussians, we derive boosted stochastic backpropagation which is regarded as the best current practice in variational inference.

We propose a novel methodology to scale gradient boosting to the mini-batch and higher dimensional setting of deep generative models with the per-batch alternation of steepest descent and line search optimization processes to avoid numerical instabilities that tend to plague mixture models in deep learning.[4]

Our promising initial results highlight the added flexibility and ease of tuning for posterior approximations even for rich nonlinear parametrization.

7.2 Future Work

As we build on this theoretical framework, we hope to establish a stronger empirical proof of the practicality and scalability of this framework on real world datasets and discriminative tasks.

Future directions would consist of rethinking of the L_2 regularization scheme suggested by [9] which does not scale well for densities.

Furthermore, as we struggled with the instability of the deep learning aspect of this research, we noticed the lack of literature linking the boundedness of network weights to the smoothness and boundedness of the top stochastic layer which would have immediate impact on the stability of similar models and algorithms.

Appendix A

Computational Complexity

Each forward pass for a single reparametrizable deep generative model costs $C_{VAE} = O(LD^2)$ where L is the total number of layers and D is the average width of the MLP layers. At each boosting iteration, we use a single Monte Carlo estimate. However, we need to compute pair-wise entropy terms that we cannot cache as in shallow models due to the neural network parametrization which enforces a unique mapping from each observation to the latent space. Therefore, at k iterations, similarly to other mixture models, the complexity would be of the order of $O(\frac{k(k-1)}{2}C_{VAE})$. However, in most practical applications, one's marginal improvement for many iterations would be so negligible that it's possible to choose a stopping criterion tailored to one's computational resources and statistical needs.

Alternatively, we can estimate the parameters of each component by taking the average over the whole dataset during training and using it instead of performing forward passes for each newinput during the training of later components or iterations.

Appendix B

Evidence Lower Bound

The Kullback-Leibler divergence is defined as:

$$KL(Q\|P) = \int_x p(x) \log \frac{p(x)}{q(x)} \tag{B.1}$$

$$\begin{aligned} KL[Q(z\|x)\|P(z\|x)] &= \mathbb{E}_Q[\log q(z\|x) - \log p(x\|z) - \log p(z)] + \log p(x) \\ \arg\min_Q KL[Q(z\|x)\|Q(z\|x)] &= \arg\min_Q \mathbb{E}_Q[\log q(z\|x) - \log p(x\|z) - \log p(z)] \\ &= \arg\min_Q KL(Q(z\|x)\|P(x,z)) \\ &= \arg\min_Q KL(Q(z\|x)\|P(z)) - \mathbb{E}_Q[\log p(x|z)] \end{aligned} \tag{B.2}$$

Notice that the KL for Gaussian distributions has an analytical form.

The RHS expectation in (B.2) is the reconstruction error or expected data log likelihood.

Throughout the paper, we use $KL(Q\|P)$ to signify the negative of thee ELBO and

our main loss function of interest:

$$KL(q\|p) = KL[Q(z\|x)\|P(z\|x)] - \log P(x) \tag{B.3}$$

Appendix C

Objective Derivation and Theoretical Guarantees

C.1 The Functional Gradient

In the case of functional gradient descent, we would like to be able to work in a generalized space of functions (such as loss functions) instead of a space of parameters. A functional $E : f \to \mathbb{R}$ is a function of functions $f \in \mathcal{H}_K$.

A functional gradient $\nabla E[f]$ is defined implicitly as the linear term of the change in a function due to a small perturbation ϵ in its inputs

$$E[f + \epsilon g] = E[f] + \epsilon \langle \nabla E[f], g \rangle + O(\epsilon^2)$$

Therefore, a functional gradient could be obtained as the limit of the gradient of a function with respect to a perturbation as the perturbation tends to 0.

$$\begin{aligned}
KL(Q+\epsilon H\|P) &= \int_x (q+\epsilon h)\log\frac{q+\epsilon h}{p} \\
&= \int_x q\log\frac{q+\epsilon h}{p} + \epsilon\int_x h\log\frac{q+\epsilon h}{p} \\
&= \int_x q\log\frac{q(1+\epsilon h/q)}{p} + \epsilon\int_x h\log\frac{q(1+\epsilon h/q)}{p} \\
&= \int_x q\log\frac{q}{p} + \int_x (q+\epsilon h)\log(1+\epsilon h/q) + \epsilon\int_x h\log\frac{q}{p} \\
&\approx KL(Q\|P) + \int_x (q+\epsilon h)\epsilon h/q + \epsilon\langle\log q-\log p, h\rangle \\
&= KL(Q\|P) + \epsilon\langle\log q-\log p, h\rangle + O(\epsilon^2) \qquad \text{(C.1)}
\end{aligned}$$

Accordingly, the functional gradient is $\nabla_{KL(Q\|P)} = \log q - \log p$

C.2 Alpha Gradients

$$\begin{aligned}
\nabla_\alpha KL((1-\alpha)F_{t-1}+\alpha Q_t\|P) &= \int_x q_t\log\frac{(1-\alpha)f_{t-1}+\alpha q_t}{p} - \int_x f_{t-1}\log\frac{(1-\alpha)f_{t-1}+\alpha q_t}{p} \\
&= \mathbb{E}_{q_t}[\log\frac{(1-\alpha)f_{t-1}+\alpha q_t}{p}] - \mathbb{E}_{f_{t-1}}[\log\frac{(1-\alpha)f_{t-1}+\alpha q_t}{p}] \\
&= \mathbb{E}_{q_t}[\log\frac{(1-\alpha)f_{t-1}+\alpha q_t}{p}] - \sum_i^t \alpha_i\mathbb{E}_{f_i}[\log\frac{(1-\alpha)f_{t-1}+\alpha q_t}{p}]
\end{aligned} \qquad \text{(C.2)}$$

$$\nabla^2_\alpha KL((1-\alpha)F_{t-1}+\alpha Q_t\|P) = \int_x \frac{[f_{t-1}-q_t]^2}{(1-\alpha)q_t+\alpha f_{t-1}} \qquad \text{(C.3)}$$

C.3 Gradient Boosting Objective

Derivation of the objective based on Friedman's [9] framework:

$$\begin{aligned}
\arg\min_q \left\| -\nabla_{KL(f_{t-1})} - \lambda(q - f_{t-1}) \right\|^2 &= \arg\min_q \int_x (-\nabla_{KL(f_{t-1})} - \lambda(q - f_{t-1}))^2 \\
&= \arg\min_q \int_x 2\nabla_{KL(f_{t-1})}\lambda(q - f_{t-1}) + \int_x \lambda^2 (q - f_{t-1})^2 \\
&= \arg\min_q \int_x \nabla_{KL(f_{t-1})} + \frac{\lambda}{2}\int_x (q - f_{t-1})^2 \\
&= \arg\min_q \int_x (\log f_{t-1} - \log p)(q - f_{t-1}) \\
&+ \frac{\lambda}{2}\int_x q^2 - q f_{t-1} \\
&= \arg\min_q \int_x q\log f_{t-1} - q\log p - f_{t-1}\log f_{t-1} + f_{t-1}\log p \\
&+ \frac{\lambda}{2}\int_x q^2 - q f_{t-1} \\
&= \arg\min_q \int_x q\log f_{t-1} - q\log q + q\log q - q\log p \\
&+ \frac{\lambda}{2}\int_x q^2 - q f_{t-1} \\
&= \arg\min_q KL(Q\|P) - KL(Q\|F_{t-1}) + \frac{\lambda}{2}\int_x q^2 - q f_{t-1}
\end{aligned} \tag{C.4}$$

C.4 Joint Convexity of the KL divergence

Similarly to other f-divergences [cite f convexity], for any $\alpha \in [0, 1]$, KL divergence is jointly convex in its arguments.

That is, for any(P_1, Q_1) and (P_2, Q_2) pairs of probability distributions over a random

variable X we have:

$$P = \alpha P_1 + (1-\alpha)P_2 \tag{C.5}$$

$$Q = \alpha Q_1 + (1-\alpha)Q_2 \tag{C.6}$$

We want to prove:

$$KL(Q|P) \leq \alpha KL(Q_1\|P_1) + (1-\alpha)KL(Q_2\|P_2) \tag{C.7}$$

Proposition: Log-Sum Inequality: If $a_1, \ , \ a_n, \ b_1, \ , \ b_n$ are non-negative numbers then:

$$\sum_{i=1}^{n} a_i \log\left(1/b_i\right) \leq \left(\sum_{i=1}^{n} a_i\right) \log\left(\frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i}\right) \tag{C.8}$$

Proof: Let

$$a_1 = \alpha q_1, a_2 = (1-\alpha)q_2 \tag{C.9}$$

$$b_1 = \alpha p_1, b_2 = (1-\alpha)p_2 \tag{C.10}$$

We then have joint convexity:

$$\begin{aligned}
KL(Q\|P) &= \int_x q \log \frac{q}{p} \\
&= \int_x \alpha q_1 + (1-\alpha)q_2 \log \frac{\alpha q_1 + (1-\alpha)q_2}{\alpha p_1 + (1-\alpha)p_2} \\
&= \int_x a_1 + a_2 \log \frac{a_1 + a_2}{b_1 + b_2} \\
&\leq \int_x a_1 \log \frac{a_1}{b_1} + a_2 \log \frac{a_2}{b_2} \\
&\leq \int_x \alpha q_1 \log \frac{\alpha q_1}{\alpha p_1} + \int_x (1-\alpha) q_2 \log \frac{(1-\alpha)q_2}{(1-\alpha)p_2} \\
&= \alpha KL(Q_1\|P_1) + (1-\alpha)KL(Q_2\|P_2)
\end{aligned} \tag{C.11}$$

In the special case of $P_1 = P_2$ we retrieve the following convexity property:

$$KL(Q\|P) \leq \alpha KL(Q_1\|P_1) + (1-\alpha)KL(Q_2\|P_1) \tag{C.12}$$

Which entails the following inequality (can be used for stopping condition):

$$KL(Q_1\|P) - KL(Q_2\|P) \leq \langle \nabla KL(Q_1\|P), Q_2 - Q_1 \rangle + \frac{1}{a}\|Q_2 - Q_1\|^2 \tag{C.13}$$

C.5 Strong Smoothness of the KL divergence

Let $Q_2 = Q_1 + \phi$

Well use the fact that

$$\log(1+x) \leq x \rightarrow \log(a+b) = \log a(1+\frac{b}{a}) \leq \log a + \frac{b}{a} \tag{C.14}$$

Proof:

$$\begin{aligned}
KL(Q_1\|P) - KL(Q_2\|P) &= KL(Q_1\|P) - KL(Q_1+\phi\|P) \\
&= \langle q_1+\phi, \log q_1 + \phi\rangle + \langle q_1+\phi, p\rangle - \langle q_1+\phi, p\rangle - \langle q_1, \log q_1\rangle \\
&= \langle q_1+\phi, \log q_1 + \phi\rangle - \langle q_1, \log q_1\rangle - \langle \phi, p\rangle \\
&= \left\langle q_1+\phi, \log q_1(1+\frac{\phi}{q_1})\right\rangle - \langle q_1, \log q_1\rangle - \langle \phi, p\rangle \\
&\leq \left\langle q_1+\phi, \log q_1 + \frac{\phi}{q_1}\right\rangle - \langle q_1, \log q_1\rangle - \langle \phi, p\rangle \\
&= \langle q_1, \log\phi\rangle + \left\langle \phi, \frac{\phi}{q_1}\right\rangle - \langle \phi, \log p\rangle \\
&\leq \langle \log q_1 - \log p, \phi\rangle + \frac{1}{a}\|\phi\|^2
\end{aligned} \tag{C.15}$$

Appendix D

Analytical Mixture Bounds and Practical Tricks

We focus on cross-entropies with Gaussians since the entropy can be written as a weighed sum of these expressions.

D.1 Mixture Cross-Entropy Lower Bound

Proof: Using Jensen's inequality:

$$
\begin{aligned}
-\int_x \mathcal{N}(x;\mu_q,\Sigma_q)\log\sum_{i=1}^{t-1}\alpha_i\mathcal{N}(x;\mu_i,\Sigma_i) &= -\log\int_x \mathcal{N}(x;\mu_q,\Sigma_q)\sum_{i=1}^{t-1}\alpha_i\mathcal{N}(x;\mu_i,\Sigma_i) \\
&\geq -\log\sum_{i=1}^{t-1}\alpha_i\int_x \mathcal{N}(x;\mu_q,\Sigma_q)\mathcal{N}(x;\mu_i,\Sigma_i) \\
&\geq -\log\sum_{i=1}^{t-1}\alpha_i\int_x \mathcal{N}(\mu_q;\mu_i,\Sigma_i+\Sigma_q) \quad \text{(D.1)}
\end{aligned}
$$

D.2 Mixture Cross-Entropy Upper Bound

$\forall\ k \leq t-1$ we have:

$$-\int_x \mathcal{N}(x;\mu_t,\Sigma_t)\log\sum_{i=1}^{t-1}\alpha_i\mathcal{N}(x;\mu_i,\Sigma_i) \leq -\log\alpha_k - \int_x \mathcal{N}(x;\mu_t,\Sigma_t)\log\mathcal{N}(x;\mu_k,\Sigma_k) \tag{D.2}$$

Proof:

$$\begin{aligned}
-\int_x \mathcal{N}(x;\mu_t,\Sigma_t)\log\sum_{i=1}^{t-1}\alpha_i\mathcal{N}(x;\mu_i,\Sigma_i) &= -\int_x \mathcal{N}(x;\mu_t,\Sigma_t)\log\left(\alpha_k\mathcal{N}(x;\mu_k,\Sigma_k)\cdot(1+\epsilon_k)\right) \\
&= -\int_x \mathcal{N}(x;\mu_t,\Sigma_t)[\log\alpha_k\mathcal{N}(x;\mu_k,\Sigma_k) + \log(1+\epsilon_k)] \\
&\leq -\int_x \mathcal{N}(x;\mu_t,\Sigma_t)[\log\alpha_k\mathcal{N}(x;\mu_k,\Sigma_k)] \\
&= \min_k CrossEntropy(\mathcal{N}(x;\mu_t,\Sigma_t),\mathcal{N}(x;\mu_k,\Sigma_k)) \\
&\quad -\log\alpha_k
\end{aligned}$$

With ϵ_i defined as follows:

$$\epsilon_i = \frac{\sum_{i\neq j=1}^{t-1}\alpha_j\mathcal{N}(x;\mu_j,\Sigma_j)}{\alpha_i\mathcal{N}(x;\mu_i,\Sigma_i)} \tag{D.3}$$

Notice that $\log(1+\epsilon_i)$ is always positive. Thus, we retrieve the upper bound.

For a tighter upper bound, we pick the component k from the mixture that minimizes the upper bound

(the cross-entropy and the log mixing coefficient).

D.3 Mixture Likelihood and LogSumExp

$$\begin{aligned}
\int_{n=1}^{N} \log p(x_n|\alpha_n, \mu_n, \sigma_n) &= \sum_{n=1}^{N} \log \sum_{i=1}^{t} \alpha_i \mathcal{N}(x_n|\mu_{i,n}, \sigma_{i,n}) \\
&= \sum_{n=1}^{N} \log \sum_{i=1}^{t} e^{\log \alpha_i \mathcal{N}(x_n|\mu_{i,n}, \sigma_{i,n})} \qquad \text{(D.4)} \\
&= \sum_{n=1}^{N} \log \sum_{i=1}^{t} e^{\log \alpha_i + \log \mathcal{N}(x_n|\mu_{i,n}, \sigma_{i,n})} \qquad \text{(D.5)}
\end{aligned}$$

A log-sum-exp operation is known to encounter numerical stability. Therefore we use the log-sum-exp

trick to subtract the maximum value before taking the log-sum-exp:

$$\begin{aligned}
x_{max} &= \arg\max_x \log \sum_i e^{x_i} \\
\log \sum_i e^{x_i} &= \log e^{x_{max}} \sum_i e^{x_i - x_{max}} \qquad \text{(D.6)} \\
&= x_{max} + \log \sum_i e^{x_i - x_{max}} \qquad \text{(D.7)}
\end{aligned}$$

Appendix E

Useful Gaussian Identities

E.1 L_2 Norm Identity for Gaussian Distributions

$$\int_x q_t^2 - q_t f_{t-1} = \mathbb{E}_{q_t}[q_t - f_{t-1}] \tag{E.1}$$

$$\begin{aligned}\int_x q_t^2 - q_t f_{t-1} &= \int \mathcal{N}(x; \mu_t, \Sigma_t) * \mathcal{N}(x; \mu_t, \Sigma_t) - \int \mathcal{N}(x; \mu_t, \Sigma_t) * \sum_{i=1}^{t-1} \alpha_i \mathcal{N}(x; \mu_i, \Sigma_i) \\ &= \int \mathcal{N}(\mu_t; \mu_t, 2 * \Sigma_t) - \sum_{i=1}^{t-1} \alpha_i \int \mathcal{N}(\mu_t; \mu_i, \Sigma_t + \Sigma_i)\end{aligned} \tag{E.2}$$

E.2 Gaussian Cross-Entropy

Let $Q = \mathcal{N}(x; \mu_1, \Sigma_1)$ and $P = \mathcal{N}(x; \mu_2, \Sigma_2)$ and D is the number of latent dimensions.

$$
\begin{aligned}
-\int_x \mathcal{N}(x; \mu_1, \Sigma_1) \log \mathcal{N}(x; \mu_1, \Sigma_1) &= -H[Q] - KL(Q\|P) \\
&= \frac{1}{2}\left[D \log (2\pi) + \log |\Sigma_2| + Tr\left(\frac{\Sigma_1}{\Sigma_2}\right) + \frac{(\mu_2 - \mu_1)^2}{\Sigma_2}\right]
\end{aligned}
\tag{E.3}
$$

E.3 Analytical KL

$$
\begin{aligned}
KL[\mathcal{N}(\mu(x), \Sigma(x))\|\mathcal{N}(0, 1)] &= \frac{1}{2}\left(\sum_d \Sigma(x) + \sum_d \mu(X)^T \mu(X) - \sum_d 1 - \log \prod_d \Sigma(x)\right) \\
&= \frac{1}{2}\left(\sum_d \Sigma(x) + \sum_d \mu(X)^T \mu(X) - \sum_d 1 - \sum_d \log \Sigma(x)\right) \\
&= \frac{1}{2}\left(\text{tr}(\Sigma(X)) + \mu(X)^T \mu(X) - k - \log \det(\Sigma(X))\right) \\
&= \frac{1}{2}\sum_d \left(\sigma_i^2(x) + \mu_i(X)^2 - 1 - \log \sigma_i^2(x)\right) \text{ for diagonal Gaussians}
\end{aligned}
\tag{E.4}
$$

In general:

$$
KL(\mathcal{N}(\mu_1, \Sigma_1)\|\mathcal{N}(\mu_2, \Sigma_2)) = \frac{1}{2}\left[tr(\Sigma_2^{-1}\Sigma_1) + \log \frac{|\Sigma_2|}{|\Sigma_1|} + (\mu_2 - \mu_1)^T \Sigma_2^{-1}(\mu_2 - \mu_1) - D\right] \tag{E.5}
$$

E.4 Gaussian Mixture Statistics

The mean and covariance of a Gaussian mixture can be computed as follows:

$$\mu_t = \mathbb{E}_{f_t}[x] = \sum_i^t \alpha_i \mu_i \tag{E.6}$$

$$\Sigma_t = Cov_{f_t}[x] = \sum_i^t \alpha_i \Sigma_i - \alpha_i \left(\mu_i - \mu_t\right)\left(\mu_i - \mu_t\right)^T \tag{E.7}$$

$$= \sum_i^t \alpha_i \sigma_i^2 + \sum_i \alpha_i \mu_i^2 - (\sum_i \alpha_i \mu_i)^2 \text{for diagonal Gaussians} \tag{E.8}$$

Bibliography

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] Christopher M Bishop, Neil Lawrence, Tommi Jaakkola, and Michael I Jordan. Approximating posterior distributions in belief networks using mixtures. *Advances in neural information processing systems*, pages 416–422, 1998.

[3] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, (just-accepted), 2017.

[4] Axel Brando. Mixture density networks (mdn) for distribution and uncertainty estimation, 2017. GitHub repository with a collection of Jupyter notebooks intended to solve a lot of problems related to MDN.

[5] Leo Breiman. Arcing the edge. Technical report, Technical Report 486, Statistics Department, University of California at Berkeley, 1997.

[6] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

[7] Xi Chen, Diederik P Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731*, 2016.

[8] Nat Dilokthanakul, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016.

[9] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[10] Yarin Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

[11] Samuel Gershman and Noah Goodman. Amortized inference in probabilistic reasoning. In *CogSci*, 2014.

[12] Samuel Gershman, Matt Hoffman, and David Blei. Nonparametric variational inference. *arXiv preprint arXiv:1206.4665*, 2012.

[13] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*, volume 9, pages 249–256, 2010.

[14] Aditya Grover and Stefano Ermon. Boosted generative models. *arXiv preprint arXiv:1702.08484*, 2017.

[15] Fangjian Guo, Xiangyu Wang, Kai Fan, Tamara Broderick, and David B Dunson. Boosting variational inference. *arXiv preprint arXiv:1611.05559*, 2016.

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[17] Tommi S Jaakkola and Michael I Jordan. Improving the mean field approximation via the use of mixture distributions. In *Learning in graphical models*, pages 163–173. Springer, 1998.

[18] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, pages 4743–4751, 2016.

[20] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[21] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, page 201611835, 2017.

[22] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[23] Yann LeCun, Corinna Cortes, and Christopher JC Burges. The mnist database of handwritten digits, 1998.

[24] Jonathan Q Li and Andrew R Barron. Mixture density estimation. In *NIPS*, volume 12, pages 279–285, 1999.

[25] Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. Auxiliary deep generative models. *arXiv preprint arXiv:1602.05473*, 2016.

[26] J Steve Marron and Matt P Wand. Exact mean integrated squared error. *The Annals of Statistics*, pages 712–736, 1992.

[27] Andrew C Miller, Nicholas Foti, and Ryan P Adams. Variational boosting: Iteratively refining posterior approximations. *arXiv preprint arXiv:1611.06585*, 2016.

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

[29] Eric Nalisnick, Lars Hertel, and Padhraic Smyth. Approximate inference for deep latent gaussian mixtures. In *NIPS Workshop on Bayesian Deep Learning*, 2016.

[30] Rajesh Ranganath, Dustin Tran, and David Blei. Hierarchical variational models. In *International Conference on Machine Learning*, pages 324–333, 2016.

[31] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.

[32] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

[33] Tim Salimans, Diederik Kingma, and Max Welling. Markov chain monte carlo and variational inference: Bridging the gap. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1218–1226, 2015.

[34] Iulian V Serban, II Ororbia, G Alexander, Joelle Pineau, and Aaron Courville. Multi-modal variational encoder-decoders. *arXiv preprint arXiv:1612.00377*, 2016.

[35] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. In *Advances in Neural Information Processing Systems*, pages 3738–3746, 2016.

[36] Ilya Tolstikhin, Sylvain Gelly, Olivier Bousquet, Carl-Johann Simon-Gabriel, and Bernhard Schölkopf. Adagan: Boosting generative models. *arXiv preprint arXiv:1701.02386*, 2017.

[37] Tong Zhang. Sequential greedy approximation for certain convex optimization problems. *IEEE Transactions on Information Theory*, 49(3):682–691, 2003.

[38] O Zobay et al. Variational bayesian inference with gaussian-mixture approximations. *Electronic Journal of Statistics*, 8(1):355–389, 2014.