Semi-unsupervised Learning using Deep Generative Models

Matthew Willetts [1,2], **Aiden Doherty**[1], **Stephen Roberts**[1,2], **Chris Holmes**[1,2]
[1] University of Oxford
[2] The Alan Turing Institute

Abstract

We introduce 'semi-unsupervised learning', a problem regime related to transfer learning and zero/few shot learning where, in the training data, some classes are sparsely labelled and others entirely unlabelled. Models able to learn from training data of this type are potentially of great use as many real-world datasets are like this. Here we demonstrate a new deep generative model for classification in this regime. Our model, a Gaussian mixture deep generative model, demonstrates superior semi-unsupervised classification performance on MNIST to model M2 from Kingma and Welling (2014).

1 Introduction

While developing machine learning solutions, the amount of unlabelled data is typically much larger than the amount of labelled data. Further, there is selection bias: the labelled data is often from a biased sample of the overall data distribution. Rare class categories might be entirely unobserved in the labelled dataset, only appearing in unlabelled data.

Thus we are interested in the case where an unlabelled instance of data could be from one of the sparsely-labelled classes or from an entirely-unlabelled class. We call this 'semi-unsupervised learning'. Here we are jointly performing semi-supervised learning on sparsely-labelled classes, and unsupervised learning on completely unlabelled classes. We give a deep generative model [1, 2] that can solve this problem.

Semi-unsupervised learning has similarities to some varieties of zero-shot learning (ZSL), where deep generative models have been of interest [3], but in ZSL one has access to auxiliary side information (commonly an 'attribute vector') for data at training time, which we do not. So our regime is equivalent to transductive generalised ZSL, but with no side information [4]. It also has similarities to transfer learning. In Cook et al.'s terms [5], 'semi-unsupervised learning' is related to uninformed semi-supervised transductive transfer learning but here we use our source and target information jointly, and our discrete label space can either be the same or different for our labelled and unlabelled data. We show our model's utility for MNIST image data classification.[1]

2 Deep Generative Models

2.1 Variational Auto-Encoder

In a deep generative model, the parameters of the distributions within a probabilistic graphical model are themselves parameterised by neural networks. The simplest is a variational autoencoder

[1] Note: related work-in-progress, with a focus on application to human activity recognition, is in the NeurIPS ML4Health Workshop 2018, titled 'Semi-unsupervised Learning of Human Activity using Deep Generative Models'.

Third workshop on Bayesian Deep Learning (NeurIPS 2018), Montréal, Canada.

[1, 2], the deep version of factor analysis. Here there is a continuous unobserved latent z and observed data x. The joint probability is $p_\theta(x, z) = p_\theta(x|z)p(z)$ with $p(z) = \mathcal{N}(0, \mathbb{I})$ and $p_\theta(x|z) = \mathcal{N}(\mu_\theta(z), \Sigma_\theta(z))$ where $\mu_\theta(z), \Sigma_\theta(z)$ are each parameterised by neural networks with parameters θ. As exact inference for $p(z|x)$ is intractable, it is standard to perform stochastic amortised variational inference to obtain an approximation $q(z|x)$ to the true posterior.

For a VAE, introduce a recognition network $q_\phi(z|x) = \mathcal{N}(\mu_\phi(x), \Sigma_\phi(x))$ (where $\mu_\phi(z), \Sigma_\phi(z)$ are neural networks with parameters θ). Through joint optimisation over $\{\theta, \phi\}$ using stochastic gradient descent we aim to find the point-estimates of the parameters $\{\theta, \phi\}$ that maximises the evidence lower bound $\mathcal{L}(x) = -\operatorname{KL}_z(q_\phi(z|x)||p_\theta(x, z))$. For the expectation over $z \sim q_\phi(z|x, y)$ in $\mathcal{L}(x)$ we take Monte Carlo (MC) samples. To take derivatives through these samples wrt θ, ϕ use the 'reparameterisation trick', rewriting a sample from a Gaussian as a deterministic function of sample from $\mathcal{N}(0, \mathbb{I})$:

$$z \sim \mathcal{N}(\mu, \sigma^2) \iff \epsilon \sim \mathcal{N}(0, \mathbb{I}), z = \mu + \sigma \cdot \epsilon \tag{1}$$

thus we can differentiate a sample w.r.t. μ, σ^2, so we can differentiate our MC approximation w.r.t θ, ϕ.

2.2 Semi-supervised Learning with Deep Generative Models

To perform semi-supervised classification with a deep generative model, introduce a discrete class variable y into the generative model and into the recognition networks. There will be two evidence lower bounds for the model, one where y is a latent variable to be inferred: $\mathcal{L}(x) = -\operatorname{KL}_{z,y}(q_\phi(z, y|x)||p_\theta(x, y, z))$ and one where y is observed: $\mathcal{L}(x, y) = -\operatorname{KL}_z(q_\phi(z|x, y)||p_\theta(x, y, z))$.

In this work we build on the M2 model developed by Kingma and Welling (2014) [6]. Here $p_\theta(x, y, z) = p_\theta(x|y, z)p(y)p(z)$ and $q_\phi(z, y|x) = q_\phi(z|y, x)q_\phi(y|x)$, $q_\phi(y|x) = \text{Cat}(\pi_\phi(x))$ and $q_\phi(z|x, y) = \mathcal{N}(\mu_\phi(x, y), \Sigma_\phi(x, y))$. $p(y)$ is the discrete prior on y. Via simple manipulation one can show $\mathcal{L}(x) = \sum_y [q_\phi(y|x)\,\mathcal{L}_\ell(x, y)] + \mathcal{H}(q_\phi(y|x))$. Note that $q_\phi(y|x)$, which is to be our trained classifier at the end, only appears in $\mathcal{L}(x)$, so it would only be trained on unlabelled data. To remedy this, motivated by considering a Dirichlet hyperprior on $p(y)$, they add to the loss the cross entropy between the true label and $q_\phi(y|x)$, weighted by a factor α. So the overall objective for model M2 with unlabelled data D_u and labelled data D_ℓ is the sum of the evidence lower bounds for all data and this classification loss:

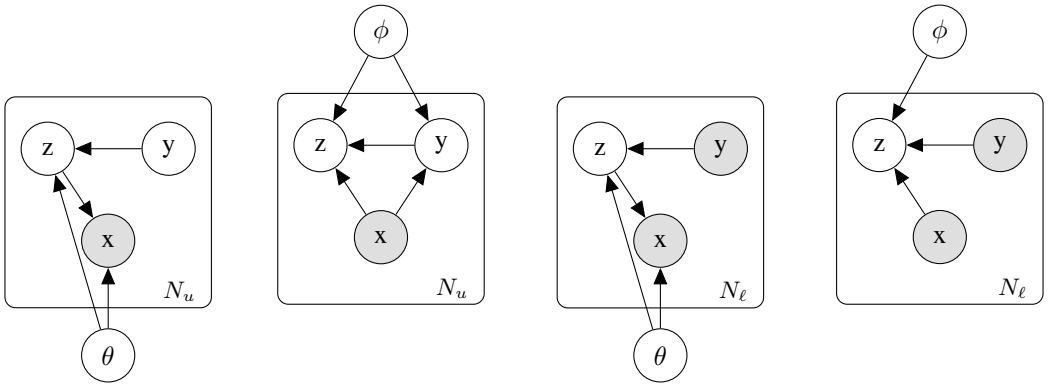
$$\mathcal{L}(D_u, D_\ell) = \sum_{x_u \sim D_u} \mathcal{L}(x_u) + \sum_{(x_\ell, y_\ell) \sim D_\ell} \Big[\mathcal{L}(x_\ell, y_\ell) + \alpha(-\log q_\phi(y_\ell|x_\ell)) \Big] \tag{2}$$

2.3 Posterior Collapse

This model has been demonstrated in the semi-supervised case [6], but when there is no label data at all, when we are just optimising $\sum_{x_u \sim D_u} \mathcal{L}(x_u)$, the model can fail to learn an informative distribution for $q_\phi(y|x)$ (see similar effect in [7], and this phenomena is well studied for the continuous latent variable z: [8, 9, 10, 11]). y can either collapse to the prior $p(y)$, or it maps every datapoint to one class. Either way the model reduces to something very similar to a standard VAE with no y variable. This happens when the encoder and decoder are high enough in capacity to obtain a locally optimal value of the evidence lower bound without using the class label. Thus, if one wishes to use high-capacity neural networks it is necessary to adjust the model in some way.

2.4 Our model - a Gaussian mixture deep generative model

Given M2's inability to consistently learn y in the semi-unsupervised case, here we propose a change to the generative structure to ameliorate posterior collapse in y. This is to enable us to learn with a mixture of semi-supervised and unsupervised classes. Many deep generative models have been proposed for semi-supervised learning, such as [12, 13] and for unsupervised learning [7, 8, 14], but none have dealt with posterior collapse in y so as to perform semi-unsupervised learning. We note that [15] proposes a large model class covering combinations of graphical models with neural networks parameterising them where inference is done using message passing in some parts of the model and gradient descent methods for the rest; here we use gradient descent exclusively.



(a) Generative Model, y latent

(b) Recognition Model, y latent

(c) Generative Model, y obs.

(d) Recognition Model, y obs.

Figure 1: Representation of our DGM as a probabilistic graphical model, for data x, partially observed class y, continuous latent z, θ, ϕ. Figures (a,c) shows the generative model $p_\theta(x|z)p_\theta(z|y)p(y)$ with y latent and observed. Figures (b,d) shows the variational approximate posterior $q_\phi(z, y|x)$ with y latent and observed.

We propose a deep generative model: a Gaussian mixture version of a variational auto-encoder, inspired by Kingma et al.'s M2 [6] and the GMM-VAE [7]. Rather than having the same distribution $p(z)$ for all classes as in M2, we condition on y to obtain a mixture of gaussians in z space. Our model, which we call a Gaussian-mixture deep generative model, or GM-DGM, is simpler than the GMM-VAE, having only one continuous latent variable. We perform semi-unsupervised classification with this model, and also compare performance with M2. Note that our model can also be trained unsupervised as well. The generative model for the data is:

$$p_\theta(x, y, z) = p_\theta(x|z)p_\theta(z|y)p(y) \tag{3}$$

$$p(y) = \text{Cat}(\pi) \tag{4}$$

$$p_\theta(z|y) = \mathcal{N}(\mu_\theta(y), \sigma^2_\theta(y)) \tag{5}$$

$$p_\theta(x|z) = \mathcal{N}(\mu_\theta(z), \sigma^2_\theta(z)) \text{ or } \mathcal{B}(\mu_\theta(z)) \tag{6}$$

We then perform amortised stochastic variational inference, with variational distributions as before for M2. See Fig. (1) for a graphical representation of our model.

Like M2, the evidence lower bound for the GM-DGM has two forms, one for if the data is labelled and one if it is not:

$$\mathcal{L}(x, y) = \mathbb{E}_{q_\phi(z|x,y)}[\log \frac{p_\theta(x|z)p_\theta(z|y)p_\theta(y)}{q_\phi(z|x, y)}] \tag{7}$$

$$= \mathbb{E}_{q_\phi(z|x,y)}[\log p_\theta(x|z)] - \text{KL}\left(q_\phi(z|x, y)||p_\theta(z|y)\right) + \log p_\theta(y) \tag{8}$$

$$\mathcal{L}(x) = \mathbb{E}_{q_\phi(z,y|x)}[\log \frac{p_\theta(x|z)p_\theta(z|y)p_\theta(y)}{q_\phi(z|x, y)q_\phi(y|x)}] \tag{9}$$

$$= \sum_y q_\phi(y|x)\big[\,\mathbb{E}_{q_\phi(z|x,y)}[\log p_\theta(x|z)] - \text{KL}\left(q_\phi(z|x, y)||p_\theta(z|y)\right)\big] - \text{KL}\left(q_\phi(y|x)||p_\theta(y)\right) \tag{10}$$

and the total objective is as in Eq. (2).

3 Experiments

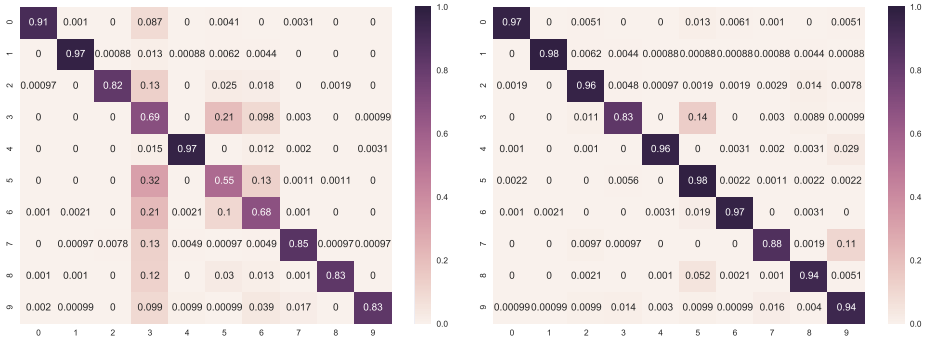
3.1 Model Implementation and MNIST results

All networks are small MLPs, 2-4 layers with 500 hidden units per layer and RELU activations. z is 100 dimensional. The same network architectures were used for networks with the same inputs and

outputs. Our code is based on the template code associated with Gordon & Hernandez-Lobato (2017) [16]. Training was done using Adam [17]. Kernel initialisation was Glorot-Normal and weights were regularised via a Gaussian prior as in [6]. [2]

Here we trained the both the GM-DGM and M2 with digits $0, 1, 2, 8, 9$ semi-supervised with 100 labels, and digits $3, 4, 5, 6, 7$ entirely unsupervised. We augmented y with 5 extra classes to learn into in addition to the 5 vacated classes. $p(y)$ was equal to 1/10 for the 5 semi-supervised classes and 1/20 for each of the 10 unsupervised classes.

To be clear, during training we are learning the classes of the digits $3, 4, 5, 6, 7$ in an unsupervised manner within our model, leveraging the latent space z which is jointly learnt with the labelled data and unlabelled data for digits $0, 1, 2, 8, 9$. We do not make use of a single labelled data point for classes $3, 4, 5, 6, 7$ in these experiments at training time. To evaluate our model at test time we observe which slots in our discrete y space that labelled examples of all classes $0 - 9$ are classified into. We attributed the learnt, unsupervised classes to the most common class within it at test time: i.e. we are performing a 'cluster-and-label' procedure. From this we then calculate accuracy. See Table (1) for overall results for 10 runs and Fig. (2) for the resulting confusion matrices from the best of 10 runs.



(a) M2 (b) Our Model

Figure 2: Best of 10 runs confusion matrices for a) M2 - accuracy 0.82 - and b) Our model - 0.94 - when trained on MNIST with 100 labelled examples each for digits 0, 1, 2, 8, 9 and digits 3, 4, 5, 6, 7 entirely unsupervised.

Model:	M2	GM-DGM
Accuracy on semi-supervised classes % (SD)	86.3 (2.2)	93.7 (1.9)
Accuracy on unsupervised classes % (SD)	63.4 (8.3)	87.4 (4.5)
Accuracy overall % (SD)	74.3 (4.4)	90.7 (2.0)

Table 1: Table showing the different performance over 10 runs for semi-unsupervised learning for both models. Trained on MNIST with 100 labelled examples each for digits 0, 1, 2, 8, 9 and digits 3, 4, 5, 6, 7 entirely unsupervised.

4 Conclusion

We show that our model, the GM-DGM, can perform better than Kingma et al's M2 [6] in semi-unsupervised learning on MNIST. y and z can be thought of as separating out class and style information about data x. Our model, through having a mixture of Gaussians in z space is a suitable choice of model when different classes in data might have different stylistic information for different classes. Its generative structure ameliorates optimisation challenges associated with VAEs with a discrete latent variable. This is work in progress, the next steps are to apply these methods, with more flexible and powerful parameterisations of the parameters of the distributions, to more advanced data sets.

[2] Code accompanying the paper is available at: github.com/MatthewWilletts/GM-DGM.

References

[1] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. *Proceedings of the 31nd International Conference on Machine Learning (ICML)*, 2014.

[2] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[3] Karl Weiss, Taghi M Khoshgoftaar, and Ding Ding Wang. A survey of transfer learning. *Journal of Big Data*, 3(1), 2016.

[4] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-Shot Learning - A Comprehensive Evaluation of the Good, the Bad and the Ugly, 2018.

[5] Diane Cook, Kyle D Feuz, and Narayanan C Krishnan. Transfer learning for activity recognition: A survey. *Knowledge and Information Systems*, 36(3):537–556, 2013.

[6] Diederik P Kingma, Danilo J Rezende, Shakir Mohamed, and Max Welling. Semi-Supervised Learning with Deep Generative Models. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[7] Nat Dilokthanakul, Pedro A M Mediano, Marta Garnelo, Matthew C H Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep Unsupervised Clustering with Gaussian Mixture VAE. *CoRR*, 2017.

[8] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance Weighted Autoencoders. *arXiv preprint*, 1509.00519, 2015.

[9] 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 (NIPS)*, 2016.

[10] Lars Maaløe, Marco Fraccaro, and Ole Winther. Semi-Supervised Generation with Cluster-aware Generative Models. In *ICML Workshop on Principled Approaches to Deep Learning*, 2017.

[11] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. How to Train Deep Variational Autoencoders and Probabilistic Ladder Networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[12] Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. Auxiliary Deep Generative Models. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 2016.

[13] Lars Maaløe, Marco Fraccaro, and Ole Winther. Semi-Supervised Generation with Cluster-aware Generative Models. *CoRR*, 2017.

[14] Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improving Variational Inference with Inverse Autoregressive Flow. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[15] Matthew James Johnson, David Duvenaud, Alexander B Wiltschko, Sandeep R Datta, and Ryan P Adams. Composing Graphical Models with Neural Networks for Structured Representations and Fast Inference. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[16] Jonathan Gordon and José Miguel Hernández-Lobato. Bayesian Semisupervised Learning with Deep Generative Models. *ICML Workshop on Principled Approaches to Deep Learning*, 2017.

[17] Diederik P Kingma and Jimmy Lei Ba. Adam: A Method for Stochastic Optimisation. *ICLR*, 2015.