

# Semi-supervised Learning with Deep Generative Models

Kingma et. al. (2014)

Tyler Brown

CS 7180

# Motivating Question

How can we model data of increasing size when obtaining label information is difficult?

## High-level Answer

We can estimate missing label information by using a probabilistic model.

# Most Relevant Previous Work

Pitelis, N., Russell, C., and Agapito, L. (2014). Semi-supervised learning using an unsupervised atlas. *In Proceedings of the European Conference on Machine Learning (ECML)*, volume LNCS 8725, pages 565 – 580.

- ▶ Observing that high-dimensional datasets often lie on or near *manifolds* of locally low rank can help avoid the *curse of dimensionality*
- ▶ Experiments show how using unlabelled data to learn the underlying manifold improves classifier accuracy when trained on limited labelled data
  1. **Unsupervised learning of the underlying manifold:** Approximate the manifold of data on the original space by fitting an atlas of low-dimensional overlapping affine charts.
  2. **Supervised training of an SVM:** Proposed a new family of Mercer Kernels for SVM-based supervised learning which uses soft-assignment of datapoints to the underlying low-dimensional affine charts to generate the kernels

# Specifying the Probabilistic Model for Missing Labels

Kingma et. al. (2014)

- ▶ Data appears as pairs  $(\mathbf{X}, \mathbf{Y}) = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$  with the  $i$ -th observation  $x_i \in \mathbb{R}^D$  and a corresponding class label  $y_i \in \{1, \dots, L\}$ 
  - ▶ Each pair of observations  $(x_i, y_i)$  has a corresponding latent variable  $z_i$
  - ▶ Empirical distribution over the labelled and unlabelled subsets is referred to as  $\tilde{p}_l(\mathbf{x}, y)$  and  $\tilde{p}_u(\mathbf{x})$
- ▶ We can estimate  $y_i$  for  $x_i$  in distribution  $\tilde{p}_u(\mathbf{x})$  by finding the maximum probability of  $p(y_i)$  by using a set of features related to  $z_i$  and a predictive model
  1. **Latent-feature discriminative model (M1)**
  2. **Generative semi-supervised model (M2)**
  3. **Stacked generative semi-supervised model (M1+M2)**

## Bayes Rule is used when specifying M1 & M2

$$\begin{aligned}p(x, y) &= p(x)p(y|x) \\ &= p(y)p(x|y) \\ p(x|y) &= \frac{p(x)p(y|x)}{p(y)}\end{aligned}$$

for models M1 <sup>1</sup>,  $p(z|x)$ , and M2 <sup>2</sup>;  $p(y|x)$

---

<sup>1</sup>Kingma et. al. (2014) equation (1)

<sup>2</sup>Kingma et. al. (2014) equation (2)

## (M1) Latent-feature discriminative model

$$y \Leftarrow p(z|x) = \frac{p(z)p(x|z)}{p(x)}$$

where

$p(z) = \mathcal{N}(z|0, I)$       Gaussian distribution of  $z$  given a missing label  $y$

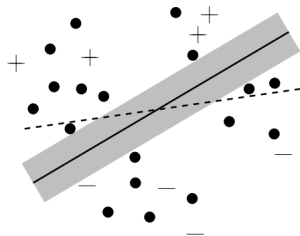
$p(x|z) = f(x; z, \theta)$       likelihood function, parameters  $\theta$  of a set of  $z$

$p(x) = \tilde{p}_u(x)$       unlabelled subset of  $x_i \in \mathbb{R}^D$

Kingma et. al. (2014) eq. (1)

## (M1) Predicting Class Labels $y$

Approximate samples from the posterior distribution over the latent variables  $p(z|x)$  are used as features to train a classifier that predicts class labels  $y$



(transductive) SVM<sup>3</sup> finds the largest margin w.r.t. the training **and** the test vectors

---

<sup>3</sup>See Figure 6.2 from Chapelle, O., B. Schölkopf, and A. Zien.  
"Semi-Supervised Learning." (2006).



## (M2) Generative semi-supervised model

$$p(y|\mathbf{x}) = \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})} \approx \frac{p_{\theta}(\mathbf{x}|y, \mathbf{z})p(y)}{p(\mathbf{x})}$$

where

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

multinomial distribution,  $y$  can be latent

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})$$

Gaussian distribution of  $\mathbf{z}$  when missing  $y$

$$p_{\theta}(\mathbf{x}|y, \mathbf{z}) = f(\mathbf{x}; y, \mathbf{z}, \theta)$$

likelihood function, nonlinear parameters

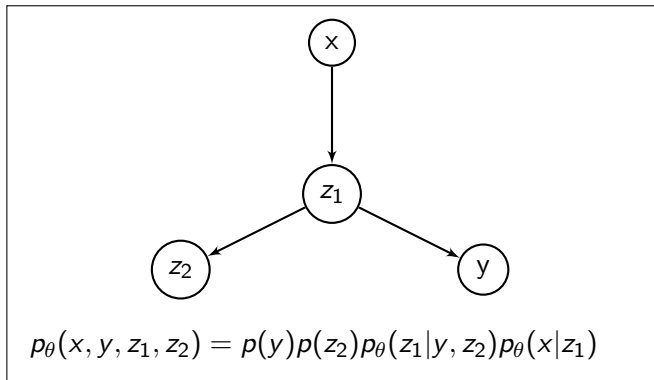
$$p(\mathbf{x})$$

all  $\mathbf{x}$  in dist. of real numbers;  $\mathbf{x} \in \mathbb{R}^D$

# Stacked generative semi-supervised model (M1 + M2)

Combine M1 and M2

1. Learn a new latent representation  $z_1$  from M1
2. Use embeddings from  $z_1$  instead of raw data  $x$ , to create a generative semi-supervised model M2



## Scaling Up: Lower Bound Objective

Lower Bound Objective<sup>4</sup>: computation of the exact posterior distribution is intractable for models M1 and M2

$$\text{M1: } q_{\phi}(z|x) = \mathcal{N}(z|u_{\phi}(x), \text{diag}(\sigma_{\phi}^2(\mathbf{x}))) \quad (3)$$

$$\begin{aligned} \text{M2: } q_{\phi}(\mathbf{z}|y, \mathbf{x}) &= \mathcal{N}(\mathbf{z}|\mu_{\phi}(y, \mathbf{x}), \text{diag}(\sigma_{\phi}^2(\mathbf{x}))); \\ q_{\phi}(y|\mathbf{z}) &= \text{Cat}(y|\pi_{\phi}(\mathbf{x})), \end{aligned} \quad (4)$$

where

|  |                                  |
|--|----------------------------------|
| $\sigma_{\phi}(x)$                               | vector of standard deviations    |
| $\pi_{\phi}(x)$                                  | probability vector               |
| $\mu_{\phi}(x), \sigma_{\phi}(x), \pi_{\phi}(x)$ | Maximum likelihood Priors (MLPs) |

---

<sup>4</sup>Kingma et. al. (2014) equations (3), (4)

# Scaling Up: M1 Model Objective

$$\log p_{\theta}(x) \geq \mathbb{E}_{q_{\phi}(z|x)}[\log p_{\theta}(x|z)] - KL[q_{\phi}(z|x)||p_{\theta}(z)] = -\mathcal{J}(x)$$

Equation 5

## Scaling Up: M2 Model Objective

When  $y_i$  is observed for the  $(x_i, y_i)$  data pair, extend from M1

$$\begin{aligned}\log p_{\theta}(x) &\geq \mathbb{E}_{q_{\phi}(z|x,y)}[\log p_{\theta}(x|y,z) + \log p_{\theta}(y) + \log p(z) - q_{\theta}(z|x,y)] \\ &= -\mathcal{L}(x,y)\end{aligned}$$

In the case where  $y_i$  is missing,

$$\begin{aligned}\log p_{\theta}(x) &\geq \mathbb{E}_{q_{\phi}(y,z|x)}[\log p_{\theta}(x|y,z) + \log p_{\theta}(y) \\ &\quad + \log p(z) - \log q_{\phi}(y,z|x)] \\ &= \sum_y q_{\phi}(y|x)(-\mathcal{L}(x,y)) + \mathcal{H}(q_{\phi}(y|x)) \\ &= -\mathcal{U}(x)\end{aligned}$$

The bound on the marginal likelihood for the entire dataset is now<sup>5</sup>

$$\mathcal{J} = \sum_{(x,y) \sim \tilde{p}_l} \mathcal{L}(x,y) + \sum_{x \sim \tilde{p}_u} \mathcal{U}(x)$$

---

<sup>5</sup>See Kingma et. al. (2014) equations 6-9

# Optimization Techniques

- ▶ Not using the EM algorithm, that's interesting, why
  - ▶ perform efficient joint inference that's easy to scale (pg. 7)
- ▶ Using AdaGrad

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

Kingma et. al. (2014) equation 11

# Optimization Algorithms

---

**Algorithm 1** Learning in model M1

---

```
while generativeTraining() do
   $\mathcal{D} \leftarrow \text{getRandomMiniBatch}()$ 
   $\mathbf{z}_i \sim q_\phi(\mathbf{z}_i | \mathbf{x}_i) \quad \forall \mathbf{x}_i \in \mathcal{D}$ 
   $\mathcal{J} \leftarrow \sum_n \mathcal{J}(\mathbf{x}_i)$ 
   $(\mathbf{g}_\theta, \mathbf{g}_\phi) \leftarrow (\frac{\partial \mathcal{J}}{\partial \theta}, \frac{\partial \mathcal{J}}{\partial \phi})$ 
   $(\theta, \phi) \leftarrow (\theta, \phi) + \Gamma(\mathbf{g}_\theta, \mathbf{g}_\phi)$ 
end while
while discriminativeTraining() do
   $\mathcal{D} \leftarrow \text{getLabeledRandomMiniBatch}()$ 
   $\mathbf{z}_i \sim q_\phi(\mathbf{z}_i | \mathbf{x}_i) \quad \forall \{\mathbf{x}_i, y_i\} \in \mathcal{D}$ 
   $\text{trainClassifier}(\{\mathbf{z}_i, y_i\})$ 
end while
```

---

---

**Algorithm 2** Learning in model M2

---

```
while training() do
   $\mathcal{D} \leftarrow \text{getRandomMiniBatch}()$ 
   $y_i \sim q_\phi(y_i | \mathbf{x}_i) \quad \forall \{\mathbf{x}_i, y_i\} \notin \mathcal{O}$ 
   $\mathbf{z}_i \sim q_\phi(\mathbf{z}_i | y_i, \mathbf{x}_i)$ 
   $\mathcal{J}^\alpha \leftarrow \text{eq. (9)}$ 
   $(\mathbf{g}_\theta, \mathbf{g}_\phi) \leftarrow (\frac{\partial \mathcal{L}^\alpha}{\partial \theta}, \frac{\partial \mathcal{L}^\alpha}{\partial \phi})$ 
   $(\theta, \phi) \leftarrow (\theta, \phi) + \Gamma(\mathbf{g}_\theta, \mathbf{g}_\phi)$ 
end while
```

---

# Results: Benchmark Classification

Table 1: Benchmark results of semi-supervised classification on MNIST with few labels.

| $N$  | NN    | CNN   | TSVM  | CAE   | MTC   | AtlasRBF            | M1+TSVM              | M2                   | M1+M2                      |
|------|-------|-------|-------|-------|-------|---------------------|----------------------|----------------------|----------------------------|
| 100  | 25.81 | 22.98 | 16.81 | 13.47 | 12.03 | 8.10 ( $\pm 0.95$ ) | 11.82 ( $\pm 0.25$ ) | 11.97 ( $\pm 1.71$ ) | <b>3.33</b> ( $\pm 0.14$ ) |
| 600  | 11.44 | 7.68  | 6.16  | 6.3   | 5.13  | –                   | 5.72 ( $\pm 0.049$ ) | 4.94 ( $\pm 0.13$ )  | <b>2.59</b> ( $\pm 0.05$ ) |
| 1000 | 10.7  | 6.45  | 5.38  | 4.77  | 3.64  | 3.68 ( $\pm 0.12$ ) | 4.24 ( $\pm 0.07$ )  | 3.60 ( $\pm 0.56$ )  | <b>2.40</b> ( $\pm 0.02$ ) |
| 3000 | 6.04  | 3.35  | 3.45  | 3.22  | 2.57  | –                   | 3.49 ( $\pm 0.04$ )  | 3.92 ( $\pm 0.63$ )  | <b>2.18</b> ( $\pm 0.04$ ) |



# Results: Image Classification

Table 2: Semi-supervised classification on the SVHN dataset with 1000 labels.

| KNN                     | TSVM                    | M1+KNN                  | M1+TSVM                 | M1+M2                          |
|-------------------------|-------------------------|-------------------------|-------------------------|--------------------------------|
| 77.93<br>( $\pm 0.08$ ) | 66.55<br>( $\pm 0.10$ ) | 65.63<br>( $\pm 0.15$ ) | 54.33<br>( $\pm 0.11$ ) | <b>36.02</b><br>( $\pm 0.10$ ) |

Table 3: Semi-supervised classification on the NORB dataset with 1000 labels.

| KNN                     | TSVM                    | M1+KNN                  | M1+TSVM                        |
|-------------------------|-------------------------|-------------------------|--------------------------------|
| 78.71<br>( $\pm 0.02$ ) | 26.00<br>( $\pm 0.06$ ) | 65.39<br>( $\pm 0.09$ ) | <b>18.79</b><br>( $\pm 0.05$ ) |

# Discussion

You can use this model for a few things  
TODO: just summarize their discussion

**Any questions? Thanks!**