Deep Hybrid Models: Bridging Discriminative and Generative Approaches

Volodymyr Kuleshov and Stefano Ermon

Department of Computer Science
Stanford University

August 2017

Overview

Discriminative vs Generative Models

Consider the task of predicting labels $y \in \mathcal{X}$ from features $x \in \mathcal{X}$.

Discriminative vs Generative Models

Consider the task of predicting labels $y \in \mathcal{X}$ from features $x \in \mathcal{X}$.

Generative Models

A generative model p specifies a joint probability $p(x, y)$ over both x and y.

Example: Naive Bayes

- Provides a richer prior
- Answers general queries (e.g. imputing features x)

Discriminative vs Generative Models

Consider the task of predicting labels $y \in \mathcal{X}$ from features $x \in \mathcal{X}$.

Generative Models

A generative model p specifies a joint probability $p(x, y)$ over both x and y.

Example: Naive Bayes

- Provides a richer prior
- Answers general queries (e.g. imputing features x)

Discriminative Models

A discriminative model p specifies a conditional probability $p(y|x)$ over y, given an x.

Example: Logistic regression.

- Focus on prediction; fewer modeling assumptions
- Lower asymptotic error

It well well-known that the decision boundary of both Naive Bayes and logistic regression has the form

$$\log \frac{p(y=1|x)}{p(y=0|x)} = b^T x + b_0.$$

It well well-known that the decision boundary of both Naive Bayes and logistic regression has the form

$$\log \frac{p(y=1|x)}{p(y=0|x)} = b^T x + b_0.$$

The difference is only training objective!

It make sense to optimize between the two.

Hybrid Models by Coupling Parameters

Hybrids Based on Coupling Parameters (McCallum et al., 2006)

1. User specifies a joint probability model $p(x, y)$.
2. We maximize the *multi-conditional likelihood*

$$\mathcal{L}(x, y) = \alpha \cdot \log p(y|x) + \beta \cdot \log p(x).$$

where $\alpha, \beta > 0$ are hyper-parameters.

- When $\alpha = \beta = 1$, we have a generative model.
- When $\beta = 0$, we have a discriminative model.

There also exists a related Bayesian coupling approach (Lasserre, Bishop, Minka, 2006)

Multi-Conditional Likelihood: Some Observations

Multi-Conditional Likelihood (McCallum et al., 2006)

Given a joint model $p(x, y)$, the multi-conditional likelihood is

$$\mathcal{L}(x, y) = \alpha \cdot \log p(y|x) + \beta \cdot \log p(x).$$

Multi-Conditional Likelihood: Some Observations

Multi-Conditional Likelihood (McCallum et al., 2006)

Given a joint model $p(x, y)$, the multi-conditional likelihood is

$$\mathcal{L}(x, y) = \alpha \cdot \log p(y|x) + \beta \cdot \log p(x).$$

Good Example: Naive Bayes

$$p(x, y) = p(x|y)p(y)$$

- $p(x) = \sum_{y \in \{0,1\}} p(x, y)$
- $p(y|x) = p(x|y)p(y)/p(x)$

Multi-Conditional Likelihood: Some Observations

Multi-Conditional Likelihood (McCallum et al., 2006)

Given a joint model $p(x, y)$, the multi-conditional likelihood is

$$\mathcal{L}(x, y) = \alpha \cdot \log p(y|x) + \beta \cdot \log p(x).$$

Good Example: Naive Bayes

$$p(x, y) = p(x|y)p(y)$$

- $p(x) = \sum_{y \in \{0,1\}} p(x, y)$
- $p(y|x) = p(x|y)p(y)/p(x)$

Bad Example: Factored $p(x, y)$

$$p(x, y) = p(y|x)p(x)$$

- $p(y|x)$ logistic regression
- $p(x)$ are word counts

Multi-Conditional Likelihood: Some Observations

Multi-Conditional Likelihood (McCallum et al., 2006)

Given a joint model $p(x, y)$, the multi-conditional likelihood is

$$\mathcal{L}(x, y) = \alpha \cdot \log p(y|x) + \beta \cdot \log p(x).$$

Good Example: Naive Bayes

$$p(x, y) = p(x|y)p(y)$$

- $p(x) = \sum_{y \in \{0,1\}} p(x, y)$
- $p(y|x) = p(x|y)p(y)/p(x)$

Bad Example: Factored $p(x, y)$

$$p(x, y) = p(y|x)p(x)$$

- $p(y|x)$ logistic regression
- $p(x)$ are word counts

Framework requires that $p(y|x)$ and $p(x)$ **share weights**!

Multi-Conditional Likelihood: Limitations

Multi-Conditional Likelihood (McCallum et al., 2006)

Given a joint model $p(x, y)$, the multi-conditional likelihood is

$$\mathcal{L}(x, y) = \alpha \cdot \log p(y|x) + \beta \cdot \log p(x).$$

Shared weights pose two types of limitations:

1. **Modeling**: limits models that we can specify (e.g. how to define $p(x, y)$ such that $p(y|x)$ is a conv. neural network)?
2. **Computational**: marginal $p(x)$, posterior $p(y|x)$ need to be tractable

A New Framework Based on Latent Variables

We couple discriminative + generative parts using *latent variables*.

1. User defines generative model with latent $z \in \mathcal{Z}$.

$$p(x, y, z) = p(y|x, z) \cdot p(x, z)$$

   The $p(y|x, z)$, $p(x, z)$ are very general; they only share the latent z, not parameters!

2. We train $p(x, y, z)$ using a multi-conditional objective

Advantages of our framework:

- Much greater modeling flexibility
- Trains complex models (incl. lat. var.) using approx. inference

Approximate Variational Inference

Consider a latent variable model $p(x, z)$ with intractable $p(x)$.

Let $q(x)$ be the data distribution and $q(z|x) \approx p(z|x)$ is an *approximate posterior* that we fit as follows.

Approximate Variational Inference

We maximize the variational lower bound on the log-likelihood:

$$\begin{aligned} \text{data log-likelihood} &= \mathbb{E}_{x \sim q(x)} \log p(x) \\ &\geq \mathbb{E}_{x \sim q(x)} \mathbb{E}_{z \sim q(z|x)} \left[\log p(x, z) - \log q(z|x)\right] \\ &= -\mathrm{KL}\left[q(x, z) || p(x, z)\right], \end{aligned}$$

Multi-Conditional Objective for Our Framework

As before, $q(x, y)$ is the data distribution and $q(z|x)$ is (learned) approximate posterior.

Generative Component

We minimize an f-divergence

$$L_G = D_f \left[q(x, z) || p(x, z) \right]$$

This encourages $q(z|x) \approx p(z|x)$ and $p(x) \approx q(x)$.

Multi-Conditional Objective for Our Framework

As before, $q(x, y)$ is the data distribution and $q(z|x)$ is (learned) approximate posterior.

Generative Component

We minimize an f-divergence

$$L_G = D_f \left[q(x, z) || p(x, z)\right]$$

This encourages $q(z|x) \approx p(z|x)$ and $p(x) \approx q(x)$.

Discriminative Component

We minimize a classification loss:

$$L_D = \mathbb{E}_{q(x,y)} \mathbb{E}_{q(z|x)} \ell\left(y, p(y|x, z)\right)$$

We may choose to minimize ℓ_2, log, hinge loss, etc.

Multi-Conditional Objective for Our Framework

As before, $q(x, y)$ is the data distribution and $q(z|x)$ is (learned) approximate posterior.

Generative Component

We minimize an f-divergence

$$L_G = D_f \left[q(x, z) || p(x, z) \right]$$

This encourages $q(z|x) \approx p(z|x)$ and $p(x) \approx q(x)$.

Discriminative Component

We minimize a classification loss:

$$L_D = \mathbb{E}_{q(x,y)} \mathbb{E}_{q(z|x)} \ell \left(y, p(y|x, z) \right)$$

We may choose to minimize ℓ_2, log, hinge loss, etc.

We fit $p(y|x, z), p(x, z), q(z|x)$ by minimizing the objective

$$L(p, q) = \alpha \cdot L_G + \beta \cdot L_D.$$

Explicit Density Models

Natural idea: bound the marginal multi-conditional log-likelihood

$$\log \int_{z \in \mathcal{Z}} p(y|x,z)^{\gamma} p(x,z) dz \geq \mathcal{L} = \text{variational lower bound.}$$

Applying the variational principle, we have our framework:

$$\mathcal{L} = \mathbb{E}_{q(z|x)} \left[\gamma \log p(y|x,z) + \log p(x,z) - \log q(z|x) \right].$$

Explicit Density Models

Natural idea: bound the marginal multi-conditional log-likelihood

$$\log \int_{z \in \mathcal{Z}} p(y|x,z)^{\gamma} p(x,z) dz \geq \mathcal{L} = \text{variational lower bound.}$$

Applying the variational principle, we have our framework:

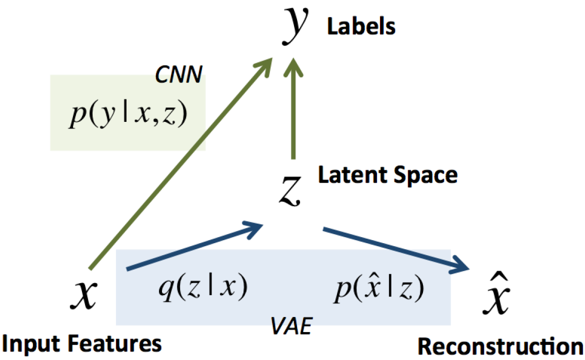
$$\mathcal{L} = \mathbb{E}_{q(z|x)} \left[\gamma \log p(y|x,z) + \log p(x,z) - \log q(z|x) \right].$$

Latent Variable Hybrid Model with Explicit Density

Suppose that $p(y|x,z)$, $p(x,z)$, $q(z|x)$ can be evaluated in closed form and have tractable gradients. We optimize

$$L_D = \text{expected log loss} \qquad L_G = \mathrm{KL}\left(q(x,z) || p(x,z)\right).$$

Deep Hybrid Models: Intuitions



- This may seen as unsupervised feature extraction
- Alternatively, we are regularizing the discriminative model

Implicit Density Models

Our framework also extends to recent GAN-based methods.

Latent Variable Hybrid Model with Implicit Density

Suppose that $p(y|x,z)$, $p(x|z)$, $q(z|x)$ are differentiable and can be sampled. We optimize

$$L_D = \text{expected log loss} \qquad L_G = \mathrm{JS}\left(q(x,z)||p(x,z)\right).$$

This amounts to parametrizing $p(x,z)$ with a generative adversarial network.

Deep Hybrid Models

Instantiating $p(x, y, z)$ with neural nets yields *deep hybrid models*.

We experiment with a particular architecture suited to vision tasks.

Generative component

Variational Autoencoder

Min. $\mathrm{KL}(q(x, z)||p(x, z))$, where

- $p(z) = \mathcal{N}(0, 1)$
- $p(x|z) = \mathcal{N}(\mu_1(z), \Sigma_1(z))$
- $q(z|x) = \mathcal{N}(\mu_2(z), \Sigma_2(z))$

Discriminative component

Convolutional Neural Network

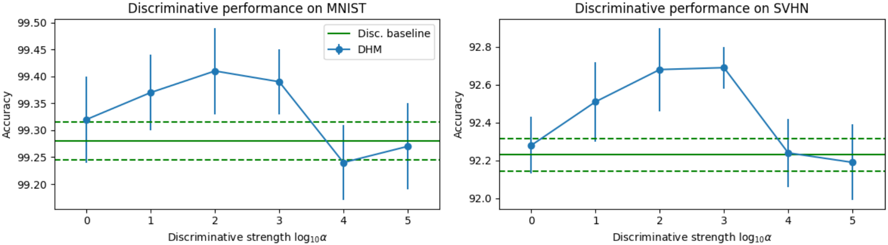
Logits ϕ from deep convolutions

- $p(y|x, z) = \mathrm{softmax}(\phi(x, z))$

All functions μ, Σ, ϕ are neural nets.

Interpolation: Discriminative Performance

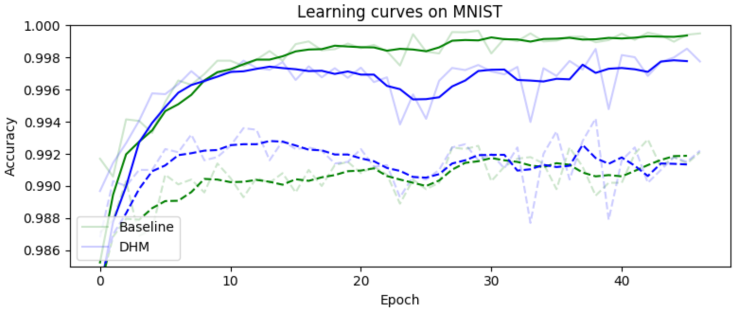
We train an explicit density model on MNIST/SVHN and vary γ.



- Adjusting discriminative strength improves performance
- Baseline assigns no weight to generative part $(\alpha = 1, \beta = 0)$

Effects of Regularization

Why does it work? Learning curves on MNIST for baseline + ours



- Our training/test error curves stay closer to each other
- This suggests a regularization effect

Semi-Supervised Learning

In semi-supervised learning, there are also two types of algorithms

Generative approaches

- Model true label y as a missing latent variable
- Semi-supervised VAE, semi-supervised GANs, etc.

Discriminative approaches

- Place decision boundary far from unlabeled data
- Transductive SVM, Entropy regularization

Our framework allows us to apply both types techniques in the same model.

Semi-Supervised Experiments: SVHN

Our framework produces improvements over state-of-the-art on semi-supervised datasets:

Method	Accuracy
VAE (Kingma et al.)	$36.02 \pm 0.10\%$
SDGM (Maaloe et al.)	$16.61 \pm 0.24\%$
Improved GAN (Salimans et al.)	$8.11 \pm 1.3\%$
ALI (Dumoulin et al.)	$7.42 \pm 0.65\%$
Π-model (Aila et al.)	$5.45 \pm 0.25\%$
Implicit HDGM (ours)	$4.45 \pm 0.35\%$

Summary

New framework for hybrid models based on latent-variable coupling. Advantages include:

- Greater flexibility when specifying the the hybrid model.
- Deals with complex models (incl. LV) using approximate inference
- Compatible with modern deep learning approaches
- Improves semi-supervised accuracy

The end

Thank you!