Deep Hybrid Models: Bridging Discriminative and Generative Approaches

Volodymyr Kuleshov and Stefano Ermon

Department of Computer Science
Stanford University

August 2017
Overview

1. **A New Framework For Hybrid Models**
   - Discriminative vs Generative Approaches
   - Hybrid Models by Coupling Parameters
   - Hybrid Models by Coupling Latent Variables

2. **An Application: Deep Hybrid Models**
   - Hybrid Models with Explicit Densities
   - Deep Hybrid Models

3. **Supervised and Semi-Supervised Experiments**
   - Supervised Experiments
   - Semi-Supervised Experiments
Consider the task of predicting labels $y \in \mathcal{Y}$ from features $x \in \mathcal{X}$. 
Consider the task of predicting labels $y \in \mathcal{Y}$ 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 $x$)
Consider the task of predicting labels $y \in \mathcal{Y}$ 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 is 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

<table>
<thead>
<tr>
<th>Hybrids Based on Coupling Parameters (McCallum et al., 2006)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. User specifies a joint probability model $p(x, y)$.</td>
</tr>
<tr>
<td>2. We maximize the multi-conditional likelihood</td>
</tr>
<tr>
<td>$\mathcal{L}(x, y) = \alpha \cdot \log p(y</td>
</tr>
<tr>
<td>where $\alpha, \beta &gt; 0$ are hyper-parameters.</td>
</tr>
</tbody>
</table>

- 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 (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
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:

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

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

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

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

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

**Discriminative Component**

We minimize a classification loss:

\[
L_D = \mathbb{E}_{q(x, y)} \mathbb{E}_{q(z|x)} \ell (y, p(y|x, z))
\]

We may choose to minimize \( \ell_2 \), 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

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

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

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

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

**Discriminative Component**

We minimize a classification loss:

\[
L_D = \mathbb{E}_{q(x, y)} \mathbb{E}_{q(z|x)} \ell (y, p(y|x, z))
\]

We may choose to minimize \( \ell_2 \), log, hinge loss, etc.
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

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

Applying the variational principle, we have our framework:

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

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

\[
L_D = \text{expected log loss} \quad L_G = \text{KL} (q(x, z) \| p(x, z)).
\]
Deep Hybrid Models: Intuitions

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

\[
L_D = \text{expected log loss} \quad \quad L_G = \text{JS} \left( q(x, z) \| p(x, z) \right).
\]

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. $\text{KL}(q(x, z) \| p(x, z))$, 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 $\phi$ from deep convolutions

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

All functions $\mu, \Sigma, \phi$ are neural nets.
We train an explicit density model on MNIST/SVHN and vary $\gamma$.

- 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:

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAE (Kingma et al.)</td>
<td>36.02 ± 0.10%</td>
</tr>
<tr>
<td>SDGM (Maaloe et al.)</td>
<td>16.61 ± 0.24%</td>
</tr>
<tr>
<td>Improved GAN (Salimans et al.)</td>
<td>8.11 ± 1.3%</td>
</tr>
<tr>
<td>ALI (Dumoulin et al.)</td>
<td>7.42 ± 0.65%</td>
</tr>
<tr>
<td>Π-model (Aila et al.)</td>
<td>5.45 ± 0.25%</td>
</tr>
<tr>
<td>Implicit HDGM (ours)</td>
<td>4.45 ± 0.35%</td>
</tr>
</tbody>
</table>
Summary

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

- Greater flexibility when specifying 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!