Generative Adversarial Active Learning

J.J. (Jia-Jie) Zhu
Boston College
Generative Adversarial Active Learning

Jia-Jie Zhu
Computer Science Department, Boston College, Chestnut Hill, MA 02467-3859, USA

José Bento
Computer Science Department, Boston College, Chestnut Hill, MA 02467-3859, USA

ZHUUV@BC.EDU
JOSE.BENTO@BC.EDU
Active learning

classify 400 instances using logistic regression

30 randomly selected
70% accuracy

30 selected using AL
90% accuracy

source: Settles ’10

pool-based active learning cycle
Generative Adversarial Active Learning

Jia-Jie Zhu
Computer Science Department, Boston College, Chestnut Hill, MA 02467-3859, USA

José Bento
Computer Science Department, Boston College, Chestnut Hill, MA 02467-3859, USA
Generative Adversarial Nets

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio
Département d’informatique et de recherche opérationnelle
Université de Montréal
Montréal, QC H3C 3J7
Intuition of GAN

The goal is to train a generator that generates “fake” data that looks as if it is “real”. (Think counterfeit bills)

- We let player 1 (discriminator D) and player 2 (generator G) play an adversarial game.
- G tries to generate “fake” data to fool D while D tries to tell “real” from “fake”.
- Both players keep getting better by playing the game. In the end, we obtain a “good” generator.

This amounts to solving the optimization problem

\[
\min_G \max_D \left\{ \mathbb{E}_{x \sim \text{data}} \log D(x) + \mathbb{E}_z \log (1 - D(G(z))) \right\}
\]

G is a generator
D(x): the probability that x comes from the real data rather than the generator
How GAN works

Main idea: match the distributions

$$\min_G \max_D \left\{ \mathbb{E}_{x \sim p_{data}} \log D(x) + \mathbb{E}_{z \sim p_z} \log (1 - D(G(z))) \right\}$$

Let $x = G(z)$, then $x \sim p_G$ the generated distribution

A lower bound for $2 \cdot \text{JSD}(p_{data}, p_G) + C$
Generative Adversarial Active Learning

Jia-Jie Zhu
Computer Science Department, Boston College, Chestnut Hill, MA 02467-3859, USA
ZHUUV@BC.EDU

José Bento
Computer Science Department, Boston College, Chestnut Hill, MA 02467-3859, USA
JOSE.BENTO@BC.EDU
Intuition of GAAL

Traditional AL (pool-based)

Can we synthesize an informative data sample on demand?

We need to generate samples that follow the same distribution as the given data

$$\min_{x \in P} \| W^T \phi(x) + b \|,$$

where $P$ is the pool of unlabeled instances, $f = W^T \phi(x) + b$ is the separating hyperplane of the SVM classifier.

Tong & Koller ’02

$$\min_{z} \| W^T \phi(G(z)) + b \|$$

$G$ is a generator in GAN
**Algorithm sketch**

**Algorithm 1: Generative Adversarial Active Learning (GAAL)**

Train generator $G$ on all unlabeled data by solving (2)

- Initialize labeled training dataset $S$ by randomly picking a small fraction of the data to label

```
repeat
    Solve optimization problem
    \[
    \min_z ||W^T \phi(G(z)) + b||
    \]
    by gradient descent
    Use the solution \( \{ z_1, z_2, \ldots \} \) and $G$ to generate instances for querying
    Label \( \{ G(z_1), G(z_2), \ldots \} \) by human oracles
    Add labeled data to the training dataset $S$ and re-train the learner, update $W, b$
```

until Labeling budget is reached
Experiments

What’s not working?
* Cats vs dogs
* Some images are garbage

Generated images
Summary of GAAL

- Generalize GAAL to other domains
- GAN is relatively unreliable as a query generator
  - We do not understand the bounds for label complexity yet
- The first work to report satisfactory results in active learning synthesis for image classification
- The first GAN application to active learning
  - The framework can be thought of as generate data that is adaptive to the current learner
  - An interesting idea. Apply similar ideas to RL/control?