Learning How to Actively Learn: A Deep Imitation Learning Approach

Ming Liu

Joint Work with: Wray Buntine and Reza Haffari
Monash University, Australia

{ming.m.liu, wray.buntine, gholamreza.haffari} @monash.edu
Roadmap

• Introduction to active learning (AL)
• Markov decision process (MDP) for agent-based AL
• Deep imitation learning to train the AL policy
• Experiments & Analysis
Roadmap

• Introduction to active learning (AL)
  • Markov decision process (MDP) for agent-based AL
  • Deep imitation learning to train the AL policy
  • Experiments & Analysis
Introduction

Raw unlabeled data points $x_1, x_2, \ldots$

Classifier

(x1,?) -> (x1,y1) -> (x2,?) -> (x2,y2) -> ... -> Oracle/Expert:

Provides labels for queries
Introduction

• At any time during the AL process, we have a current guess for the classifier

• **AL Strategy**: Query the point closest to the decision boundary
Introduction

**Warnings:**
- Not clear whether heuristics lead to optimal querying behavior
- Not clear which hard coded heuristic is good for a task at hand

 AL Heuristics \((x_1, ?)\)

\((x_1, y_1)\)

AL Heuristics \((x_2, ?)\)

\((x_2, y_2)\)

\(\cdots\)

**Oracle/Expert:**
Provides labels for queries

Classifier
Introduction

Can we learn the best active learning strategy?

Classifier

Oracle/Expert: Provides labels for queries
Roadmap

• Introduction to active learning (AL)
• Markov decision process (MDP) for agent-based AL
• Deep imitation learning to train the AL policy
• Experiments & Analysis
Agent-based Active Learning

Need to train an AL agent to tell what data to select next, given
• the previously selected data
• the pool of unlabeled data available
• the underlying classifier, learned so far

AL Agent: (x1,?)
(x1,y1)

AL Agent: (x2,?)
(x2,y2)

Classifier

Oracle/Expert: Provides labels for queries
AL Query Strategy by an Agent

Raw unlabeled data points $x_1, x_2, \ldots$

The Tutoring AL Agent & Learning Student (Classifier)

Oracle/Expert: Provides labels for queries
Agent Operates in Markov Decision Process

\[ s_1, a_1, s_2, a_2, s_3, \ldots \]

Reward: Accuracy ( )

Learn the Optimal Query Policy
Roadmap

• Introduction to active learning (AL)
• Markov decision process (MDP) for agent-based AL
• Deep imitation learning to train the AL policy
• Experiments & Analysis
• **IDEA**: Let’s train the agent based on AL simulation for a rich-data task and then transfer it to AL problem of interest

• This is **Meta-Learning**: Learning to Actively Learn
  
  • Synthesize many AL problems

• Use **Imitation/Reinforcement Learning** algorithms
Synthesizing AL Problems

\[ \mathbb{E}_{(D_{lab}, D_{unl}, D_{dev}) \sim \mathcal{D}} \left[ \mathbb{E}_{\pi \theta} \left[ \sum_{t=1}^{B} R(s_t, a_t, s_{t+1}) \right] \right] \]
Imitation Learning

- The *algorithmic oracle* gives the correct action in each world state.

- Train the *agent (policy network)* to prefer the “correct” action compared to “incorrect” ones (i.e. classification).
Algorithmic Oracle

- It computes the correct action in each world state
  - Re-train the underlying model using all possible queries/actions
  - Mark the one leading to the most accurate prediction on the evaluation set

\[
\text{argmax}_{(x_i, y_i) \in \text{Pool}} \text{Accuracy} \left( \text{Retrain(} x_i, y_i \text{, Evaluation Set) } \right)
\]

- Too slow for typical large pools of data
- IDEA: Randomly sample a subset and maximize over it
  - Leads to efficient training and effective learned policies
Imitation Learning **DAGGER**

- The collected state-action pairs are not i.i.d. hence problematic for classifier learning

- **Data Aggregation (DAGGER):** Once in a while, use the predicted action by the policy network during training (Ross et al 2011)

\[ \pi_{\tau} = \beta_{\tau} \tilde{\pi}^* + (1 - \beta_{\tau}) \hat{\pi}_{\tau} \]

- This is to make sure the policy sees **bad states** and the correct action to **recover** from them in the training time
Roadmap

• Introduction to active learning (AL)
• Markov decision process (MDP) for agent-based AL
• Deep imitation learning to train the AL policy
• Experiments & Analysis
Experiments (Task 1: text classification)

- **Sentiment Classification**: Positive/Negative sentiment of a review
  - Train the AL policy on one product, and apply to the reviews of another

- **Authorship Profiling**: Gender of the author of a tweet
  - Train the AL policy on one language, and apply to another

<table>
<thead>
<tr>
<th>src</th>
<th>tgt</th>
<th>number src/tgt</th>
<th>doc. (src/tgt) number</th>
<th>avg. len. (tokens)</th>
</tr>
</thead>
<tbody>
<tr>
<td>elec.</td>
<td>music dev.</td>
<td>27k/1k</td>
<td>35/20</td>
<td></td>
</tr>
<tr>
<td>book</td>
<td>movie</td>
<td>24k/2k</td>
<td>140/150</td>
<td></td>
</tr>
<tr>
<td>en</td>
<td>sp</td>
<td>3.6k/4.2k</td>
<td>1.15k/1.35k</td>
<td></td>
</tr>
<tr>
<td>en</td>
<td>pt</td>
<td>3.6k/1.2k</td>
<td>1.15k/1.03k</td>
<td></td>
</tr>
</tbody>
</table>
Experiments (Baseline methods)

- Random sampling
- Uncertainty-based sampling
- Diversity-based sampling
- PAL (Fang et al., 2017) : A deep reinforcement learning based approach, they designed a Q-network for stream-based AL
Experiments (Task 1: text classification)
Experiments (Task 1: text classification)

- **Direct transfer**: Initialize the classifier on the source data, without AL
- **Cold-start**: Start training the classifier from random initialization, continue training with AL agent
- **Warm-start**: Start training the classifier from the pre-trained model on the source data, continue training with AL agent
Experiments (Task 2: Named Entity Recognition)

• Data sets: CoNLL 2002/2003

<table>
<thead>
<tr>
<th>Bilingual</th>
<th>Multilingual</th>
</tr>
</thead>
<tbody>
<tr>
<td>tgt</td>
<td>tgt</td>
</tr>
<tr>
<td>src</td>
<td>src</td>
</tr>
<tr>
<td>de</td>
<td>de</td>
</tr>
<tr>
<td>en</td>
<td>en, nl, es</td>
</tr>
<tr>
<td>nl</td>
<td>nl</td>
</tr>
<tr>
<td>en</td>
<td>en, de, es</td>
</tr>
<tr>
<td>es</td>
<td>es</td>
</tr>
<tr>
<td>en</td>
<td>en, de, nl</td>
</tr>
</tbody>
</table>

Table 2: Experimental settings for cross-lingual NER, in which source language (src) is used for policy training.
Experiments (Task 2: Named Entity Recognition)
Analysis: Insight on the selected data

\[
\text{acc} = \frac{\text{total } \# \text{ of overlapped examples}}{\text{budget}}
\]

\[
\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i}
\]

<table>
<thead>
<tr>
<th></th>
<th>movie sentiment</th>
<th>gender pt</th>
<th>NER es</th>
</tr>
</thead>
<tbody>
<tr>
<td>acc Unc.</td>
<td>0.06</td>
<td>0.58</td>
<td>0.51</td>
</tr>
<tr>
<td>MRR Unc.</td>
<td>0.083</td>
<td>0.674</td>
<td>0.551</td>
</tr>
<tr>
<td>acc Div.</td>
<td>0.05</td>
<td>0.52</td>
<td>0.45</td>
</tr>
<tr>
<td>MRR Div.</td>
<td>0.057</td>
<td>0.593</td>
<td>0.530</td>
</tr>
<tr>
<td>acc PAL</td>
<td>0.15</td>
<td>0.56</td>
<td>0.52</td>
</tr>
</tbody>
</table>

We use MRR (Mean reciprocal rank) and acc to show the agreement of queried data points returned by our AL agent and other strategies.
Analysis: Sensitivity to K (size of unlabeled subset)

K: size of subset from the original unlabelled set
Analysis: $\beta$ (schedule parameter for the policy)

\[ \pi_\tau = \beta_\tau \hat{\pi}^* + (1 - \beta_\tau) \hat{\pi}_\tau \]

Options for $\beta$
- Fixed: $\beta=0.5$
- Linear: $\beta_\tau = \max(0.5, 1 - 0.01\tau)$
- Exponential: $\beta_\tau = 0.9^\tau$
- Inverse sigmoid: $\beta_\tau = \frac{5}{5 + \exp(\tau/5)}$
Related work

• **Meta learning** eg learning to learn without gradient descent by gradient descent (Chen et al 2016)

• **Stream-based** AL as MDP; learning the policy with reinforcement learning (Fang et al, 2017) suffers from the credit assignment problem (Bechman et al 2017)

• **Imitation Learning**: Learning from expert demonstrations eg (Schaal 2009, Abbeel & Ng 2004, Silver et al 2008)
Conclusion

• Use heuristics or learn an agent for the AL query strategy.
• Agent-based AL as a Markov Decision Process.
• Formulate learning AL strategies/policies as an imitation learning problem.
• Our imitation learning approach performs better than previous heuristic-based and RL-based methods.
Thanks