Learning Algorithms for Active Learning
Learning Algorithms for Active Learning

Philip Bachman* 1  Alessandro Sordoni* 1  Adam Trischler 1

Plan

● Background
  ○ Matching Networks
  ○ Active Learning
● Model
● Applications: Omniglot and MovieLens
● Critique and discussion
Background: Matching Networks (Vinyals et al. 2016)

\[ \hat{y} = \sum_{i=1}^{k} a(f'(\hat{x}), g'(x_i))y_i \]

- embedding of probe item
- embedding of example
- cosine distance (e.g.)
- label of example
Background: Matching Networks

Motivates context-sensitive encoder
Background: Matching Networks

\[ g(x_i, S) = \overrightarrow{h_i} + \overleftarrow{h_i} + g'(x_i) \]
Background: Matching Networks

Desiderata for $\hat{x}$ encoding:
- Depend on embeddings of examples, $g(S)$
- Be able to selectively ignore some examples (e.g. outliers)
- Build invariance to the order of the examples

\[
\text{attLSTM}(f'(\hat{x}), g(S), K)
\]

\[
\hat{h}_k, c_k = \text{LSTM}(f'(\hat{x}), [h_{k-1}, r_{k-1}], c_{k-1})
\]

\[
h_k = \hat{h}_k + f'(\hat{x})
\]

\[
r_{k-1} = \sum_{i=1}^{|S|} a(h_{k-1}, g(x_i))g(x_i)
\]

\[
a(h_{k-1}, g(x_i)) = \text{softmax}(h_{k-1}^Tg(x_i))
\]
Background: Active Learning

- Most real-world settings: many unlabeled examples, few labeled ones
- *Active Learning*: Model requests labels; tries to maximize both task performance and data efficiency
  - E.g. task involving medical imaging: radiologist can label scans by hand, but it’s costly
- Instead of using heuristics to select items for which to request labels, Bachman et al. use meta learning to learn an active learning strategy for a given task
Proposed Model: “Active MN”

Algorithm 1 End-to-end active learning loop (for Eq. 3)

1: # encode items in $S$ with context-sensitive encoder
2: # and encode items in $E$ with context-free encoder
3: $S = \{(x, y)\}, S_0^k = \{(x, \cdot)\}, S_0^h = \emptyset, E = \{(\hat{x}, \hat{y})\}$
4: for $t = 1 \ldots T$ do
5:   # select next instance
6:   $i \leftarrow \text{SELECT}(S_t^u, S_{t-1}^k, h_{t-1})$
7:   # read labeled instance and update controller
8:   $(x_i, y_i) \leftarrow \text{READ}(S, i)$
9:   $h_t \leftarrow \text{UPDATE}(h_{t-1}, x_i, y_i)$
10:   # update known / unknown set
11:   $S_t^k \leftarrow S_{t-1}^k \cup \{(x_i, y_i)\}$
12:   $S_t^u \leftarrow S_{t-1}^u \setminus \{(x_i, \cdot)\}$
13:   # perform fast prediction (save loss for training)
14:   $L_t^S \leftarrow \text{FAST-PRED}(S, S_t^u, S_t^k, h_t)$
15: end for
16: # perform slow prediction (save loss for training)
17: $L_T^E \leftarrow \text{SLOW-PRED}(E, S_T^u, S_T^k, h_T)$
Individual Modules

Context Free and Sensitive Encodings

- Gain context by using a bi-directional LSTM over independent encodings

Selection

- At each step $t$, places a distribution $P_t^{u}$ over all unlabeled items in $S_t^{u}$
- $P_t^{u}$ computed using a gated, linear combination of features that measure controller-item and item-item similarity

Reading

- Concatenates embedding and label for item selected, then applies linear transformation

Controller

- Input: $r_t$ from reading module, and applies LSTM update: $h_t = \text{LSTM}(h_{t-1}, r_t)$
Prediction Rewards

**Prediction Reward:** \( R(E, S_t, h_t) \equiv \sum_{(\hat{x}, \hat{y}) \in E} \log p(\hat{y}|\hat{x}, h_t, S_t) \)

**Objective:** maximize \( \frac{\mathbb{E}}{(S,E) \sim \mathcal{D}} \left[ \mathbb{E}_{\pi(S,T)} \left[ \sum_{t=1}^{T} R(E, S_t, h_t) \right] \right] \)

\[ \rightarrow \quad \frac{\mathbb{E}}{(S,E) \sim \mathcal{D}} \left[ \mathbb{E}_{\pi(S,T)} \left[ \sum_{t=1}^{T} \tilde{R}(S^u_t, S_t, h_t) + R(E, S_T, h_T) \right] \right] \]

**Fast Prediction**

- Attention-based prediction for each unlabeled item using cosine sim. to labeled items
  - Sharpened by a non-negative matching score between \( x^u_i \) and the control state
- Similarities between context-sensitive embeddings don’t change with \( t \) -> can be precomputed

**Slow Prediction**

- Modified Matching Network prediction
  - Takes into account distinction between labeled and unlabeled items
  - Conditions on active learning control state
Full Algorithm

**Algorithm 1** End-to-end active learning loop (for Eq. 3)

1. # encode items in $S$ with context-sensitive encoder
2. # and encode items in $E$ with context-free encoder
3. $S = \{(x, y)\}, S^u_0 = \{(x, \cdot)\}, S^k_0 = \emptyset, E = \{(\hat{x}, \hat{y})\}$
4. **for** $t = 1 \ldots T$ **do**
5. # select next instance
6. $i \leftarrow $ SELECT($S^u_{t-1}, S^k_{t-1}, h_{t-1}$)
7. # read labeled instance and update controller
8. $(x_i, y_i) \leftarrow $ READ($S, i$)
9. $h_t \leftarrow $ UPDATE($h_{t-1}, x_i, y_i$)
10. # update known / unknown set
11. $S^k_t \leftarrow S^k_{t-1} \cup \{(x_i, y_i)\}$
12. $S^u_t \leftarrow S^u_{t-1} \setminus \{(x_i, \cdot)\}$
13. # perform fast prediction (save loss for training)
14. $L^S_t \leftarrow $ FAST-PRED($S, S^u_t, S^k_t, h_t$)
15. **end for**
16. # perform slow prediction (save loss for training)
17. $L^E_T \leftarrow $ SLOW-PRED($E, S^u_T, S^k_T, h_T$)
Tasks

Goal: maximize some combination of task performance and data efficiency

Test model on:

- Omniglot
  - 1623 characters from 50 different alphabets
- MovieLens (bootstrapping a recommender system)
  - 20M ratings on 27K movies by 138K users
Experimental Evaluation: Omniglot Baseline Models

1. Matching Net (random)
   a. Choose samples randomly

2. Matching Net (balanced)
   a. Ensure class balance

3. Minimum-Maximum Cosine Similarity
   a. Choose items that are different
## Experimental Evaluation: Omniglot Performance

*Table 1. Results for our active learner and baselines for the $N$-way, $K$-shot classification settings.*

<table>
<thead>
<tr>
<th>Model</th>
<th>5-way</th>
<th>10-way</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1-shot</td>
<td>2-shot</td>
</tr>
<tr>
<td>Matching Net (random)</td>
<td>69.8%±0.10</td>
<td>93.1%±0.07</td>
</tr>
<tr>
<td>Matching Net (balanced)</td>
<td>97.9%±0.07</td>
<td>98.9%±0.07</td>
</tr>
<tr>
<td>Active MN</td>
<td>97.4%±0.11</td>
<td>99.0%±0.08</td>
</tr>
<tr>
<td>Min-Max-Cos</td>
<td>97.4%±0.11</td>
<td>99.3%±0.02</td>
</tr>
</tbody>
</table>
Experimental Evaluation: Data Efficiency

Omniglot Performance

MovieLens Performance
Conclusion

Introduced model that learns active learning algorithms end-to-end.

- Approaches optimistic performance estimate on Omniglot
- Outperforms baselines on MovieLens
Critique/Discussion Points

- Controller doesn’t condition its label requests on the probe item

Critique/Discussion Points

- Controller doesn’t condition its label requests on the probe item
- In Matching Networks, the embeddings of the examples don’t depend on the probe item
Critique/Discussion Points

- Active learning is useful in settings where data is expensive to label, but meta-learned active learning requires lots of labeled data for training, even if this labeled data is spread across tasks. Can you think of domains where this is / is not a realistic scenario?
Critique/Discussion Points

● Active learning is useful in settings where data is expensive to label, but meta-learned active learning requires lots of labeled data for training, even if this labeled data is spread across tasks. Can you think of domains where this is / is not a realistic scenario?

● In their ablation studies, they observed that taking out the context-sensitive encoder had no significant effect. Are there any applications where you think this encoder could be essential?

● In this work, they didn’t experiment with NLP tasks. Are there any NLP tasks you think this approach could help with?