An Adaptive Feature Selection Method for Learning-to-Enumerate Problem

Satoshi Horikawa
Chiyonosuke Nemoto
Keishi Tajima
Masaki Matsubara
Atsuyuki Morishima

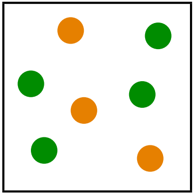
University of Tsukuba
Kyoto University

What is Learning-to-Enumerate Problem [1]?

Problem Setting

- Given a fixed set of items
- Find n instances of the target class with examining a min number of items
- Start with no classifier/training data

item set



“find 3 ●”

In this study, we also assume:

- Start with a hand-crafted noisy feature set

including both useful ones and useless ones

[1] Jörger et al., “Learning to Enumerate”, ICANN 2016

What is **Learning-to-Enumerate Problem**?

Example

- Given a corpus of news articles.
- Find 10 news that may influence the business of Toyota.
- No existing classifier, no training data,
- but we can come up with many candidate keywords:

“EV”, “hybrid”, “oil”, “Tesla”, ...

↑

a noisy feature set

What is Learning-to-Enumerate Problem?

Basic Strategy of L-to-E

Find positive instances + train a classifier in parallel

Until we find n positive instances, repeat:

1. Select an item to label based on some criteria.
2. Manually label it, and add it to the training dataset.
3. If it is a positive instance,
   add it also to the set of found items.
4. Retrain a classifier with the expanded dataset.

What is Learning-to-Enumerate Problem?

L-to-E $\neq$ Active Learning

What is common

- Label items and re-train a classifier repeatedly.

Difference

- Active Learning: to obtain a good classifier
  $\rightarrow$ label items that best improve the classifier
- L-to-E: to find positive instances
  $\rightarrow$ exploitation-exploration trade-off

Jörger et al.[1]: exploitation-only strategy works well

What is Learning-to-Enumerate Problem?

Exploitation-only Strategy for L-to-E

Until we find n positive instances, repeat:

1. Select the most likely item by the current classifier.
2. Manually label it, and add it to the training dataset.
3. If it is a positive instance,
   add it also to the set of found items.
4. Retrain a classifier with the expanded dataset.

Exploitation-only L-to-E ≈ Relevance Feedback

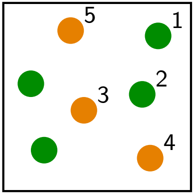
Problems to Solve

1. We assume we start with a noisy feature set.

   How can we quickly discard useless ones?

2. Drawback of exploitation-only strategy:

   The classifier is biased to clusters found earlier.



Basic Approach of Our Method AdaFeaSE

1. Binary weights to features.
   - Compare discriminative power of features.
   - Inactivate features inferior to any others.

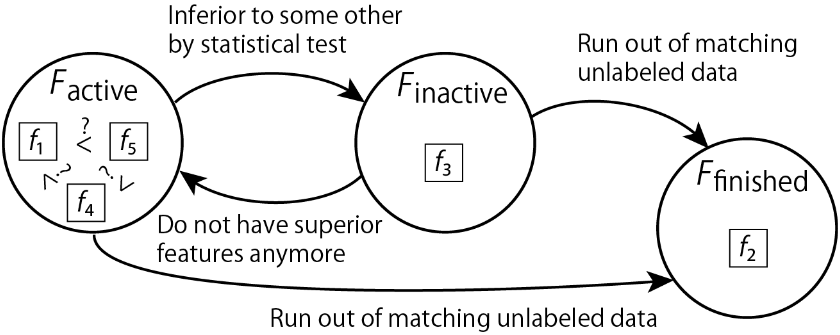
"hybrid" > "oil" → inactivate "oil" (weight=0)

2. Adaptively change the features to use.
   - When we run out of items having some feature, we re-activate features that were inferior to it.

no remaining article includes "hybrid"
→ re-activate "oil"

Details of AdaFeaSE (1/4)

State Transition of Features



Details of AdaFeaSE (2/4)

Algorithm (1/2)

Initialization of variables:

F_{active}	$=$	{all features}	active features
$F_{inactive}$	$=$	$\emptyset$	inactive features
$F_{finished}$	$=$	$\emptyset$	finished features
$X_{labeled}$	$=$	$\emptyset$	already labeled items
$X_{unlabeled}$	$=$	{all items}	remaining items
Ans	$=$	$\emptyset$	found positive items

Details of AdaFeaSE (3/4)

Algorithm (2/2)

While $|\boldsymbol{Ans}| < \boldsymbol{n}$

1. Select $\boldsymbol{x} \in \boldsymbol{X}_{\boldsymbol{unlabeled}}$ having the largest number of $\boldsymbol{f} \in \boldsymbol{F}_{\boldsymbol{active}}$
2. Label $\boldsymbol{x}$, move $\boldsymbol{x}$ from $\boldsymbol{X}_{\boldsymbol{unlabeled}}$ to $\boldsymbol{X}_{\boldsymbol{labeled}}$
3. If $\boldsymbol{positive(x)}$, add $\boldsymbol{x}$ to $\boldsymbol{Ans}$
4. Move features no remaining item has to $\boldsymbol{F}_{\boldsymbol{finished}}$
5. Reassign $\boldsymbol{f} \notin \boldsymbol{F}_{\boldsymbol{finished}}$ to $\boldsymbol{F}_{\boldsymbol{active}}$ or $\boldsymbol{F}_{\boldsymbol{inactive}}$ based on the expanded dataset $\boldsymbol{X}_{\boldsymbol{labeled}}$

Details of AdaFeaSE (4/4)

Example

active	inactive	finished
EV hybrid oil		

unlabeled items

. . . hybrid . . . EV . . .

EV, hybrid . . .

. . . EV . . . oil

. . . EV . . .

. . . oil . . . EV

labeled items

Details of AdaFeaSE (4/4)

Example

active	inactive	finished
EV hybrid oil		

unlabeled items

- . . . hybrid . . . EV . . .
- EV, hybrid . . .
- . . . EV . . . oil
- . . . EV . . .
- . . . oil . . . EV

labeled items

- ✓ . . . hybrid . . . EV . . .

Details of AdaFeaSE (4/4)

Example

active	inactive	finished
EV hybrid oil		

unlabeled items

- EV, hybrid . . .
- . . . EV . . . oil
- . . . EV . . .
- . . . oil . . . EV

labeled items

- ✓ . . . hybrid . . . EV . . .
- ✗ . . . EV . . . oil

Details of AdaFeaSE (4/4)

Example

active	inactive	finished
EV hybrid oil	oil	

unlabeled items

- EV, hybrid . . .
- . . . EV . . .
- . . . oil . . . EV

labeled items

- ✓ . . . hybrid . . . EV . . .
- ✗ . . . EV . . . oil

Details of AdaFeaSE (4/4)

Example

active	inactive	finished
EV hybrid	oil	

unlabeled items

EV, hybrid . . .

. . . EV . . .

. . . oil . . . EV

labeled items

✓ . . . hybrid . . . EV . . .

✓ EV, hybrid . . .

✗ . . . EV . . . oil

Details of AdaFeaSE (4/4)

Example

active	inactive	finished
EV hybrid	oil	hybrid

unlabeled items

... EV ...

... oil ... EV

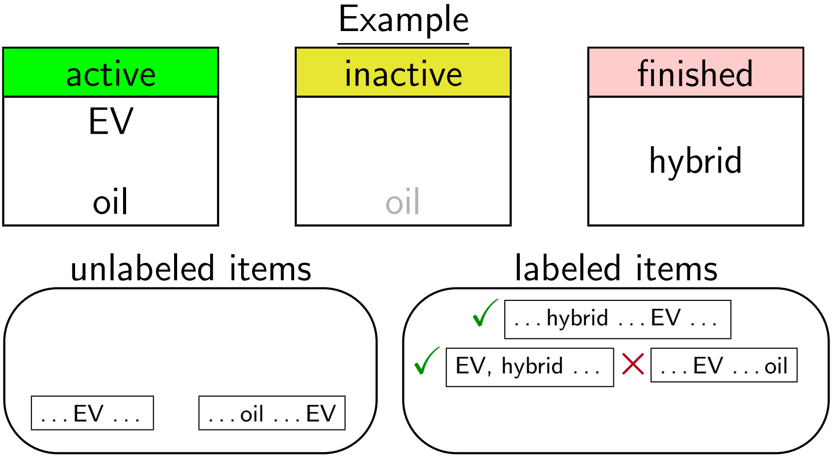
labeled items

✓ ... hybrid ... EV ...

✓ EV, hybrid ...

✗ ... EV ... oil

Details of AdaFeaSE (4/4)



Experiment

Dataset

News Corpus 6,684 news articles by Yahoo! Japan

Two Topics

1. Events influencing Toyota (79 positive instances)
2. Scandals of celebrities (252 positive instances)

Initial Feature Set (keyphrases)

- Crowdsourced
- 272 phrases for Toyota and 286 for Scandal

Experiment

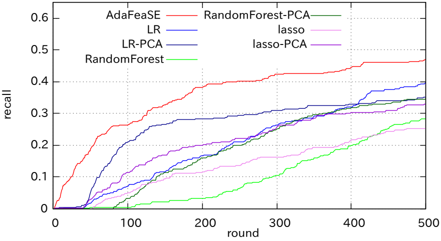
Baselines

- logistic regression (LR)
- logistic regression (LR) + PCA
- random forest (RF)
- random forest (RF) + PCA
- Lasso
- Lasso + PCA

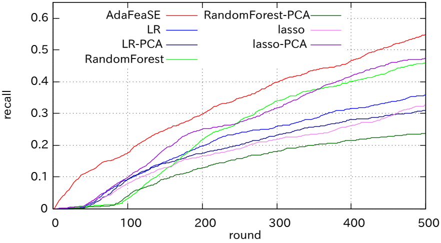
⋆ Deep learning does not work well with small training data.

Experiment

Result



Toyota



Scandal

- AdaFeaSE (red) performs best within these rounds.
- Others overtake when trained enough in later rounds.
- AdaFeaSE is quicker to find e.g. 40 articles.

Discussion

Why binary?

- Lasso, which uses regularization, also worked well.
- Binary = more aggressive regularization.
- Our method can aggressively assign weight 0 because we can re-activate it later.

Why exploitaion-only?

- The feature sets were complete enough for finding e.g., 50 items.
- Exploring given candidate features must be enough.

Key: Noisy and complete enough feature set available