Cost-sensitive Dynamic Feature Selection

He He, Hal Daumé III and Jason Eisner

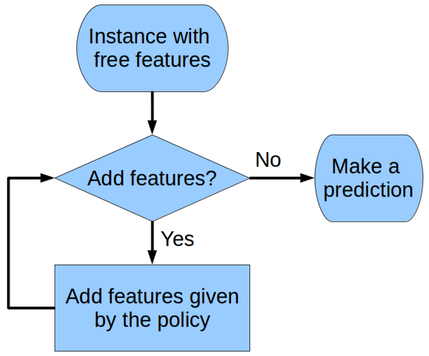
Introduction

Dynamic Feature Selection

- *Instance-specific* feature selection at *test time*
- User-specified *accuracy-cost trade-off*

Feature Selection as an MDP

Given a pretrained model and feature costs,



State s_t : selected features and their values

Action a_t : features to add and stop (make a prediction using the pretrained model with features added so far)

Policy π: map from state to action $\pi(s_t) = a_t$

Reward: $r(s_t, a_t) = \text{margin}(s_t, a_t) - \lambda \cdot \text{cost}(s_t, a_t)$

* margin: difference between score of the true label and the highest score of other labels

Imitation Learning via Classification

Oracle: demonstrate optimal actions $\pi^*(s_t) = a_t^*$

Agent: mimic the oracle's behavior $\pi(s_t) = a_t$

Training examples: $\{(\phi(s_{\pi^*}), \pi^*(s_{\pi^*}))\}$

Policy feature: $\phi(s_t)$ describing the state

Minimizing a surrogate loss $\ell(\pi, s)$ → classifier π e.g. hinge loss in SVM.

* s_{π^*} : states visited by π^*

Forward Selection Oracle

- Greedily add the feature that yields maximum reward at each step until all are selected
- Stop in the maximum-reward state
- Only available during training

Policy Features

State feature

- selected features and their values

Meta-features

- confidence score given by the pretrained classier
- change of confidence score from the previous step
- whether the prediction changes from the previous step
- cost of the current feature set
- change in cost from the previous step
- cost divided by confidence score
- guess using the current feature set

Theorem 1 (Ross & Bagnell, 2010)

Let $\mathrm{E}_{s_{\pi^*}}[\ell(s,\pi)] = \varepsilon$, then $J(\pi) \le J(\pi^*) + T^2\varepsilon$.

* $J(\pi)$: task loss (negative reward)

* T: task horizon

Why do we have qudratically increasing loss?

- Trains on states visited by the oracle only
- Ignores the difference between the oracles and the agent's state distribution

Dataset Aggregation (Dagger)

Let $\pi_1 = \pi^*$, in iteration i to N

collect training examples $D_i = \{(\phi(s_{\pi_i}), \pi^*(s_{\pi_i}))\}$

train π_{i+1} on aggregated dataset $D_1 \bigcup D_2 \bigcup \ldots \bigcup D_i$

Return the best policy on validation set.

- Run the learned policy and label examples with the oracle action
- Include the agent's distribution
- Correct mistakes

Theorem 2 (Ross *et al.*, 2011)

If $Q^*_{T-t+1}(s,\pi) - Q^*_{T-t+1}(s,\pi^*) \le u$ and N is $\tilde{O}(uT)$,

there is a policy $\pi \in \pi_{1:N}$ s.t. $J(\pi) \le J(\pi^*) + uT\varepsilon_N + O(1)$.

* $Q^{\pi'}_{T-t+1}(s,\pi)$: t-step cost of executing π in the initial state then running π'

* $\varepsilon_N = \min_{\pi \in \Pi} \frac{1}{N} \sum_{i=1}^{N} \mathrm{E}_{s_{\pi_i}}[\ell(s,\pi)]$: minimum loss in the hindsight

Dagger with Coaching

Oracle's policy can be too good to learn!

- Far from the agent's learning policy space
- Policy features are insufficient

Use kernels or more descriptive policy features, but…

large overhead !

"Hope" Action

$$\tilde{a}_t^* = \arg\max_{a \in A_t} \eta \cdot \text{score}_{\pi_i}(a) + r(s_t, a)$$

instead of the oracle action

$$a_t^* = \arg\max_{a \in A_t} r(s_t, a)$$

Coach

- Demonstrates easier-to-learn actions that the current policy prefers and has a high reward
- Approach the oracle gradually by shrinking η

Experimental Result

Baselines

Statically add features from a ranked list

- Sort by static forward selection method
- Sort by feature weight (given by the pretrained model) divided by the cost

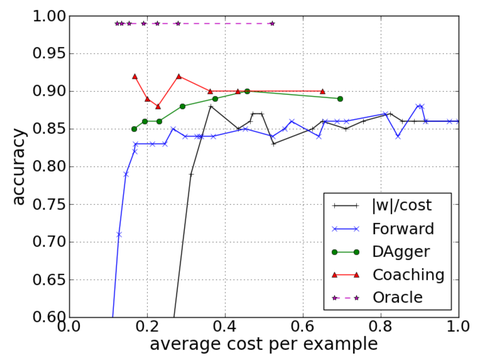
Dagger and Coaching

10 iterations

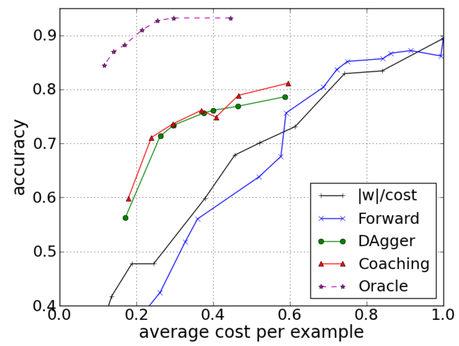
$\lambda = 0, 0.1, 0.25, 0.5, 1, 1.5, 2$

$\eta_i = e^{-(i-1)} \quad (i \in [1, N])$

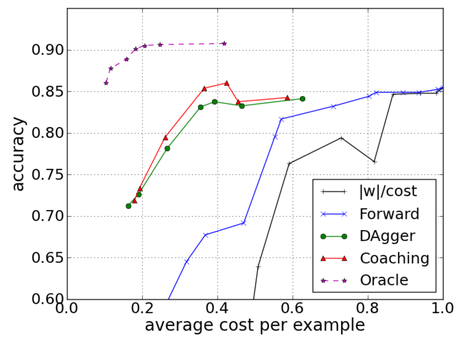
Ionosphere dataset (binary)



Digit dataset (10 classes)



Segmentation dataset (7 classes)



Conclusion and Future Work

We propose a dynamic feature selection algorithms at test time and learn the policy using imitation learning techniques. In the future, we are interested in

- learning feature weights *jointly* with dynamic feature selection policy
- including dependence between features using properties of feature templates

We would also like to explore structured prediction problems where

- policy features may require inference under features selected so far
- part of the cost need to inferred at runtime

Reference

[1] S. Ross and J. A. Bagnell. Efficient reductions for imitation learning. In AISTATS, 2010.

[2] S. Ross, G. J. Gordon and J. A. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In AISTATS, 2011.