Dynamic Feature Selection for Classification on a Budget

Sergey Karayev SERGEYK@EECS.BERKELEY.EDU
Mario J. Fritz MFRITZ@MPI-INF.MPG.DE
Trevor Darrell TREVOR@EECS.BERKELEY.EDU

1. Method

*The **test-time efficient classification problem** consists of*

- *N instances labeled with one of K labels: $\mathcal{D} = \{x_n \in \mathcal{X}, y_n \in \mathcal{Y} = \{1, \dots, K\}\}_{n=1}^{N}$.*
- *F features $\mathcal{H} = \{h_f : \mathcal{X} \mapsto \mathbb{R}^{d_f}\}_{f=1}^{F}$, with associated costs c_f.*
- *Budget-sensitive loss $\mathcal{L}_\mathcal{B}$, composed of cost budget $\mathcal{B}$ and loss function $\ell(\hat{y}, y) \mapsto \mathbb{R}$.*

*The goal is to find a **feature selection** policy $\pi(x) : \mathcal{X} \mapsto 2^\mathcal{H}$ and a **feature combination** classifier $g(\mathcal{H}_\pi) : 2^\mathcal{H} \mapsto \mathcal{Y}$ such that such that the total budget-sensitive loss $\sum \mathcal{L}_\mathcal{B}(g(\pi(x_n)), y_n)$ is minimized.*

The cost of a selected feature subset $\mathcal{H}_{\pi(x)}$ is $C_{\mathcal{H}_\pi(x)}$. The budget-sensitive loss $\mathcal{L}_\mathcal{B}$ presents a **hard budget constraint** by only accepting answers with $C_\mathcal{H} \leq \mathcal{B}$. Additionally, $\mathcal{L}_\mathcal{B}$ can be **cost-sensitive**: answers given with less cost are more valuable than costlier answers. The motivation for the latter property is ***Anytime*** performance; we should be able to stop our algorithm's execution at any time and have the best possible answer

1.1. Feature selection as an MDP.

*The **feature selection MDP** consists of the tuple $(\mathcal{S}, \mathcal{A}, T(\cdot), R(\cdot), \gamma)$:*

- ***State** $s \in \mathcal{S}$ stores the selected feature subset $\mathcal{H}_{\pi(x)}$ and their values and total cost $C_{\mathcal{H}_{\pi(x)}}$.*
- *The set of **actions** $\mathcal{A}$ is the set of features $\mathcal{H}$.*
- *The (stochastic) **state transition** distribution $T(s' \mid s, a)$ can depend on the instance x.*
- *The **reward** function $R(s, a, s') \mapsto \mathbb{R}$ is manually specified, and depends on the classifier g and the instance x.*
- *The discount γ determines amount of **lookahead** in selecting actions.*

Reward definition is given in Figure 1.

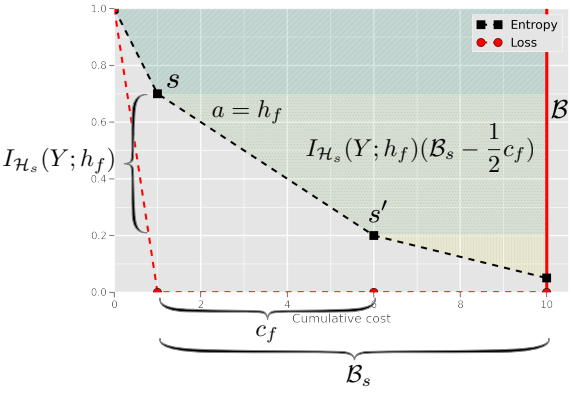

Figure 1. Definition of the reward function. We seek to maximize the total area above the entropy vs. cost curve from 0 to $\mathcal{B}$, and so define the reward of an individual action as the area of the slice of the total area that it contributes. From state s, action h leads to state s' with cost c_f. The information gain of the action $a = h_f$ is $I_{\mathcal{H}_s}(Y; h_f) = H(Y; \mathcal{H}_s) - H(Y; \mathcal{H}_s \cup h_f)$.

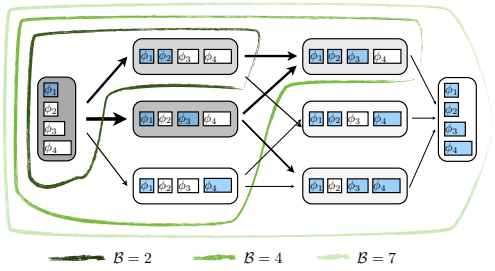

Figure 2. Illustration of the state space. The feature selection policy π induces a distribution over feature subsets, for a dataset, which is represented by the shading of the larger boxes. Not all states are reachable for a given budget $\mathcal{B}$. We show three such "budget cuts."

1.2. Learning the policy.

We learn the state feature weight vector θ by *policy iteration*. First, we gather (s, a, r, s') samples by running episodes with the current policy parameters θ_i. From these samples, $\hat{Q}(s, a)$ values are computed, and θ_{i+1} are given by L_2-regularized least squares solution

to $\hat{Q}(s,a) = \theta^T \phi(s,a)$, on all states that we have seen in training. During training, we gather samples starting from either a random feasible state, with decaying probability ϵ, or from the initial empty state otherwise.

1.3. Learning the classifier.

A **Gaussian Naive Bayes** classifier can combine an arbitrary feature subset $\mathcal{H}_\pi$, but suffers from its restrictive generative model. We found **logistic regression** to work better, but because it always uses the same length feature vector, unobserved feature values need to be imputed. We evaluate **mean** and **Gaussian** imputation.

Since the classifier depends on the distribution over states (see Figure 2) induced by the policy, and the policy training depends on the entropy of the classifier, the learning procedure is iterative, alternating between learning policy and classifier.

Note that the policy π selects some feature subsets more frequently than others. Instead of learning only one classifier g that must be robust to all observed feature subsets, we can learn several classifiers—for each of the most frequent subsets—and **match** test instances to classifier accordingly.

2. Evaluation

We evaluate the following baselines:

- **Static, greedy**: corresponds to best performance of a policy that does not observe feature values and selects actions greedily ($\gamma = 0$).
- **Static, non-myopic**: policy that does not observe values but considers future action rewards ($\gamma = 1$).
- **Dynamic, greedy**: policy that observes feature values, but selects actions greedily.

Our method is the **Dynamic, non-myopic** policy: feature values are observed, with full lookahead.

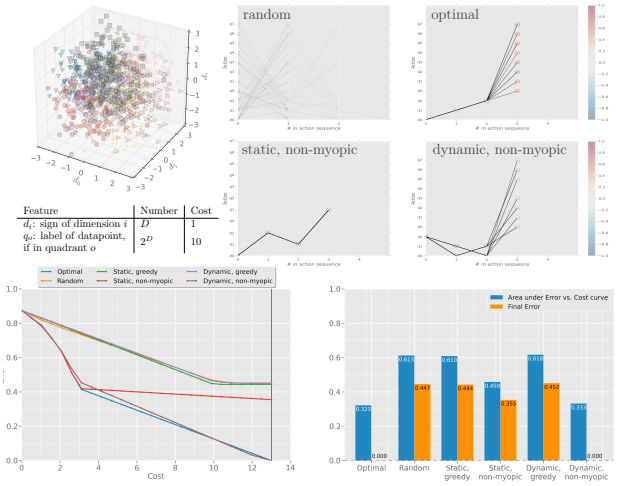

Figure 3. Evaluation on the 3-dimensional synthetic example (best viewed in color). The data is shown at top left; the sample feature trajectories of four different policies at top right. The plots in the bottom half show that we recover the known optimal policy.

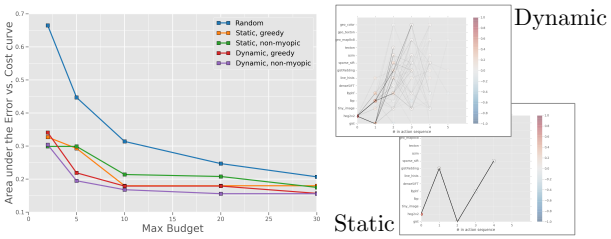

Figure 4. Results on Scenes-15 dataset (best viewed in color). 14 visual features variy in cost from 0.3 to 8 seconds, and in accuracy from 0.32 to .82. Our results on this dataset match the reported results of Active Classification (Gao & Koller, 2011) and exceed the reported results of Greedy Miser (Xu et al., 2012).

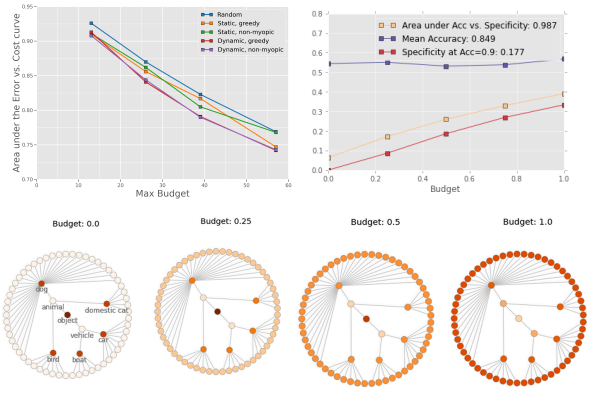

Figure 5. Results on the Imagenet 65-class subset. Note that when our method is combined with Hedging Your Bets (Deng et al., 2012), a constant accuracy can be achieved, with *specificity* of predictions increasing with the budget, as in human visual perception. (Color saturation corresponds to percentage of predictions at node.)

References

Deng, Jia, Krause, Jonathan, Berg, Alexander C, and Fei-fei, Li. Hedging Your Bets: Optimizing Accuracy-Specificity Trade-offs in Large Scale Visual Recognition. In *CVPR*, 2012.

Gao, Tianshi and Koller, Daphne. Active Classification based on Value of Classifier. In *NIPS*, 2011.

Xu, Zhixiang, Weinberger, Kilian Q, and Chapelle, Olivier. The Greedy Miser: Learning under Test-time Budgets. In *ICML*, 2012.