Cost-Sensitive Active learning in VW

Akshay Krishnamurthy

akshay@cs.umass.edu

With Alekh Agarwal, Tzu-Kuo Huang, Hal Daumé III, John Langford

Cost-Sensitive Learning

Multi-class prediction where different predictions incur different cost.

- Data (x, c) where $c(y)$ is cost for predicting label y on x.

Cost-Sensitive Learning

Multi-class prediction where different predictions incur different cost.

- Data (x, c) where $c(y)$ is cost for predicting label y on x.
- **Training**: With data (x_i, c_i)

$$g_y = \operatorname*{argmin}_{g \in \mathcal{G}} \sum_i (g(x_i) - c_i(y))^2$$

Cost-Sensitive Learning

Multi-class prediction where different predictions incur different cost.

- Data (x, c) where $c(y)$ is cost for predicting label y on x.
- **Training**: With data (x_i, c_i)

$$g_y = \operatorname*{argmin}_{g \in \mathcal{G}} \sum_i (g(x_i) - c_i(y))^2$$

- **Prediction**: On example x

$$\hat{y} = \operatorname*{argmin}_{y} g_y(x)$$

Cost-Sensitive Learning

Multi-class prediction where different predictions incur different cost.

- Data (x, c) where $c(y)$ is cost for predicting label y on x.
- **Training**: With data (x_i, c_i)

$$g_y = \operatorname*{argmin}_{g \in \mathcal{G}} \sum_i (g(x_i) - c_i(y))^2$$

- **Prediction**: On example x

$$\hat{y} = \operatorname*{argmin}_y g_y(x)$$

- **In VW:**

```
vw --csoaa k
```

Cost-Sensitive Learning

Multi-class prediction where different predictions incur different cost.

- Data (x, c) where $c(y)$ is cost for predicting label y on x.
- **Training**: With data (x_i, c_i)

$$g_y = \operatorname*{argmin}_{g \in \mathcal{G}} \sum_i (g(x_i) - c_i(y))^2$$

- **Prediction**: On example x

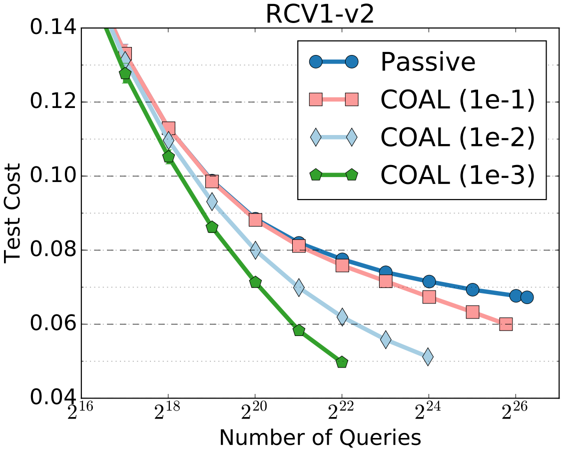
$$\hat{y} = \operatorname*{argmin}_y g_y(x)$$

- **In VW:**

```
vw --csoaa k
```

Can we do active learning here?

Yes we can!




```
vw --cs_active k --mellowness 0.01 --simulation --adax
```

Cost Overlapped Active Learning (COAL)

On each x_i

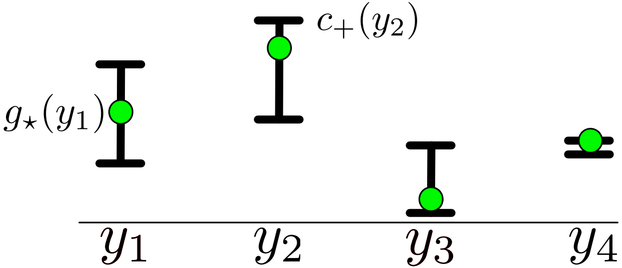
1. Compute version space $\mathcal{G}_i(y)$ of good regressors for each y.

Cost Overlapped Active Learning (COAL)

On each x_i

1. Compute version space $\mathcal{G}_i(y)$ of good regressors for each y.
2. Compute cost ranges

$$c_-(y) = \operatorname*{argmin}_{g \in \mathcal{G}_i(y)} g(x_i), \qquad c_+(y) = \operatorname*{argmax}_{g \in \mathcal{G}_i(y)} g(x_i)$$



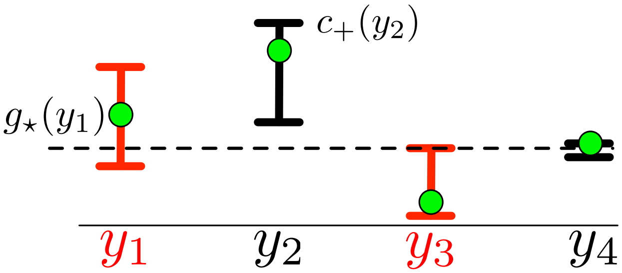
Cost Overlapped Active Learning (COAL)

On each x_i

1. Compute version space $\mathcal{G}_i(y)$ of good regressors for each y.
2. Compute cost ranges

$$c_-(y) = \operatorname*{argmin}_{g \in \mathcal{G}_i(y)} g(x_i), \qquad c_+(y) = \operatorname*{argmax}_{g \in \mathcal{G}_i(y)} g(x_i)$$

3. Query y if cost range is large and overlaps with best.



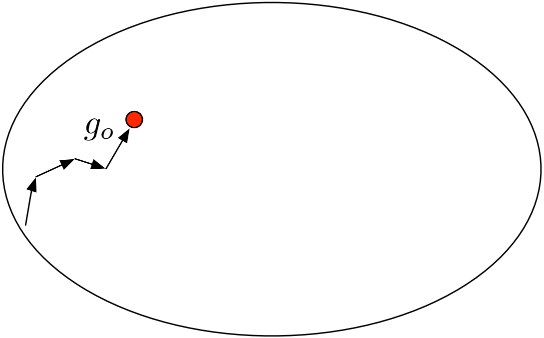
Properties

1. Guaranteed good generalization (adapts to easy data)
2. Logarithmic label complexity in favorable cases
3. In theory, polynomial time.

Approximate cost ranges

In practice, one pass, linear time.

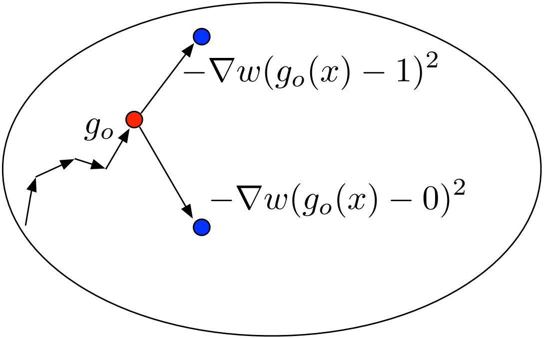
1. Use online least squares optimization.



Approximate cost ranges

In practice, one pass, linear time.

1. Use online least squares optimization.
2. Compute cost range with sensitivity analysis.
3. Look for large weight w such that with new weighted example, loss is still close



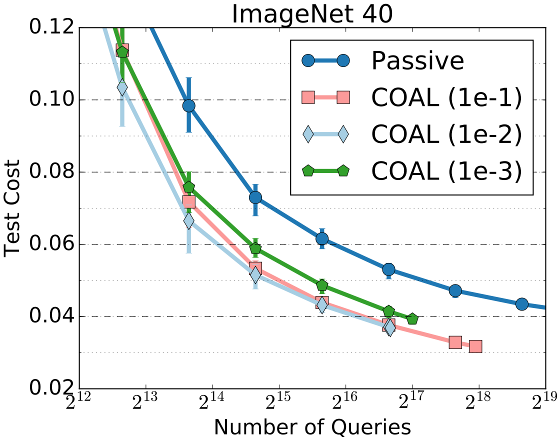
Execution

./vw –cs_active 3 -d ../test/train-sets/cs_test –cost_max 2 –mellowness 0.01 –simulation –adax
Num weight bits = 18
learning rate = 0.5
initial_t = 0
power_t = 0.5
using no cache
Reading datafile = ../test/train-sets/cs_test
num sources = 1

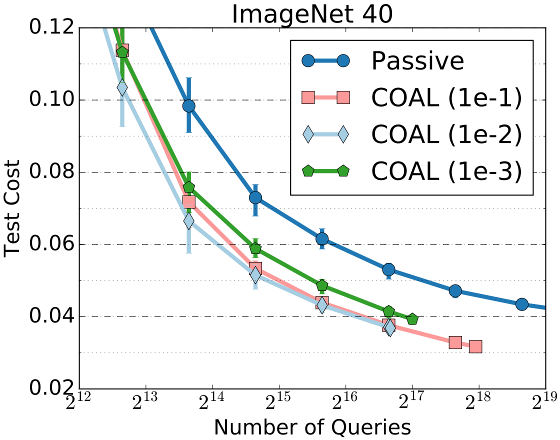
average loss	since last	example counter	example weight	current label	current predict	current features
1.000000	1.000000	1	1.0	known	1	4
0.500000	0.000000	2	2.0	known	2	4

finished run
number of examples per pass = 3
passes used = 1
weighted example sum = 3.000000
weighted label sum = 0.000000
average loss = 0.333333
total feature number = 12
total queries = 3

Experiments



Experiments



- Hierarchical classification with tree-distance cost.
- COAL gets lower test cost than passive with $\approx$ 4x fewer queries.