Tutorial on Recent Practical Vowpal Wabbit Improvements

Zhen Qin

zqin001@cs.ucr.edu

UNIVERSITY OF CALIFORNIA
UC RIVERSIDE

- Vaclav Petricek

- Lihong Li, Nikos Karampatziakis, John Langford

github SOCIAL CODING

JohnLangford / vowpal_wabbit

forked from gparker/vowpal_wabbit

Unwatch 130 | Unstar 900 | Fork 211

Microsoft® Research

Online, effective, and efficient

Heavily used in a dozen of companies (and their productions)

Simply works

In this tutorial…

Several new features added in Vowpal Wabbit

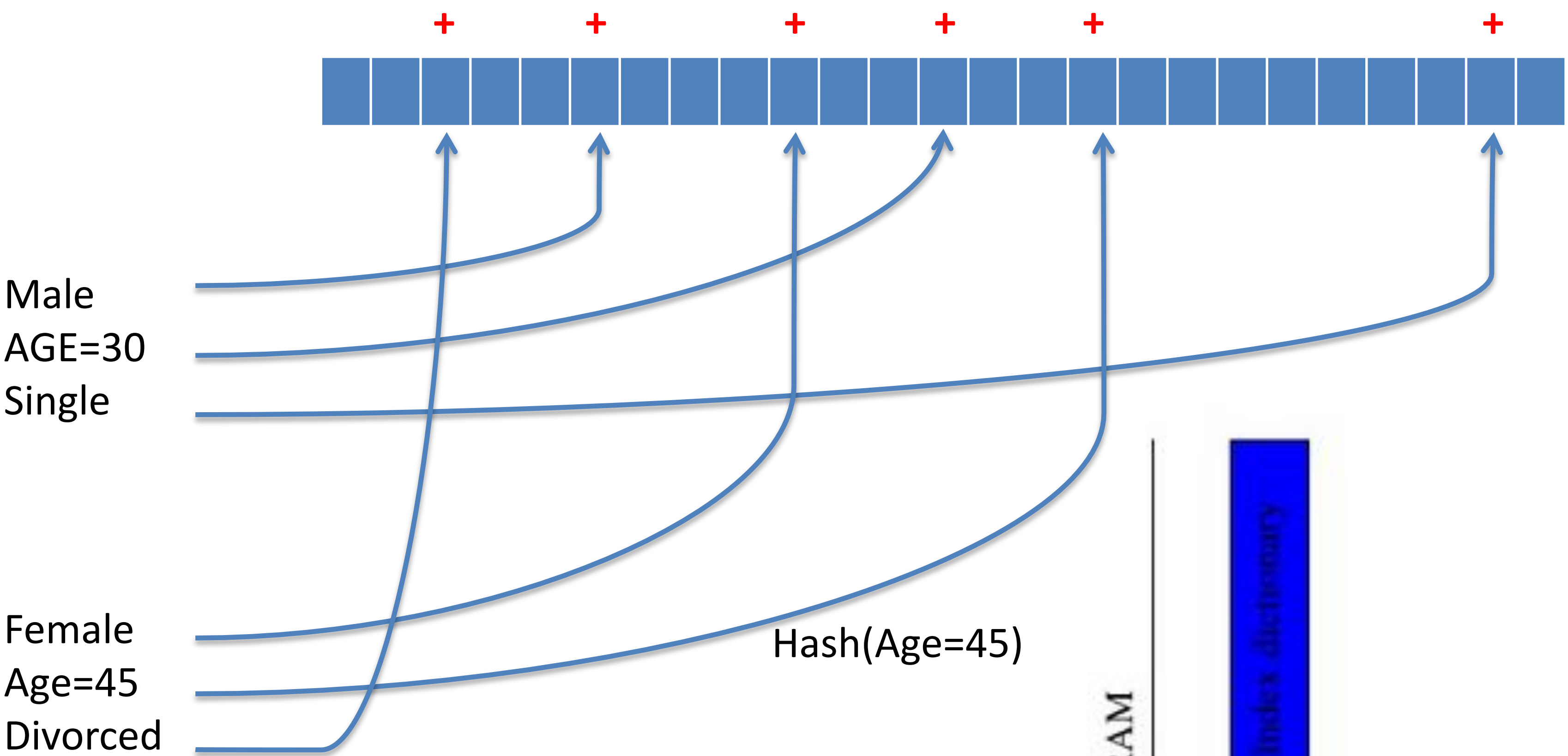
(usability, measure of generalization, new reductions)

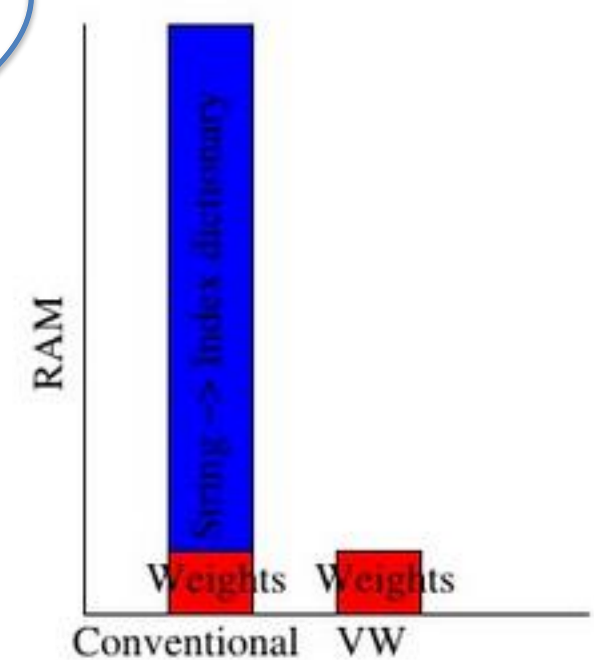
Techniques used in VW

Why (related) new features necessary

How they help and how to use them in practice

Hashing Trick





--invert_hash

#1

--readable_model	--invert_hash
2135463:0.4234	Age=45:0.4234
3462733:-0.1111	Age=30:-0.1111
1367328:0.4401	Male:0.4401
1231234:-0.0021	Female:-0.0021

NEW

--invert_hash

#1

Use existing --audit code

1. Intercept feature name and index
2. Store in map<string,int>
3. Output the readable model

NEW

--invert_hash

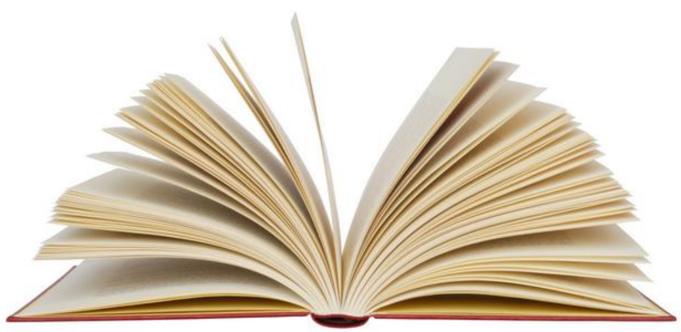
#1

vw ……. –invert_hash file.txt

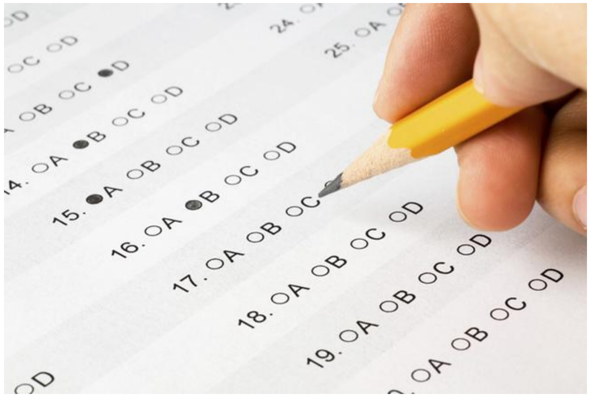
Supports gd, oaa, csoaa, wap …

Measuring model performance

It is generalization that matters



training



testing

Measuring model performance

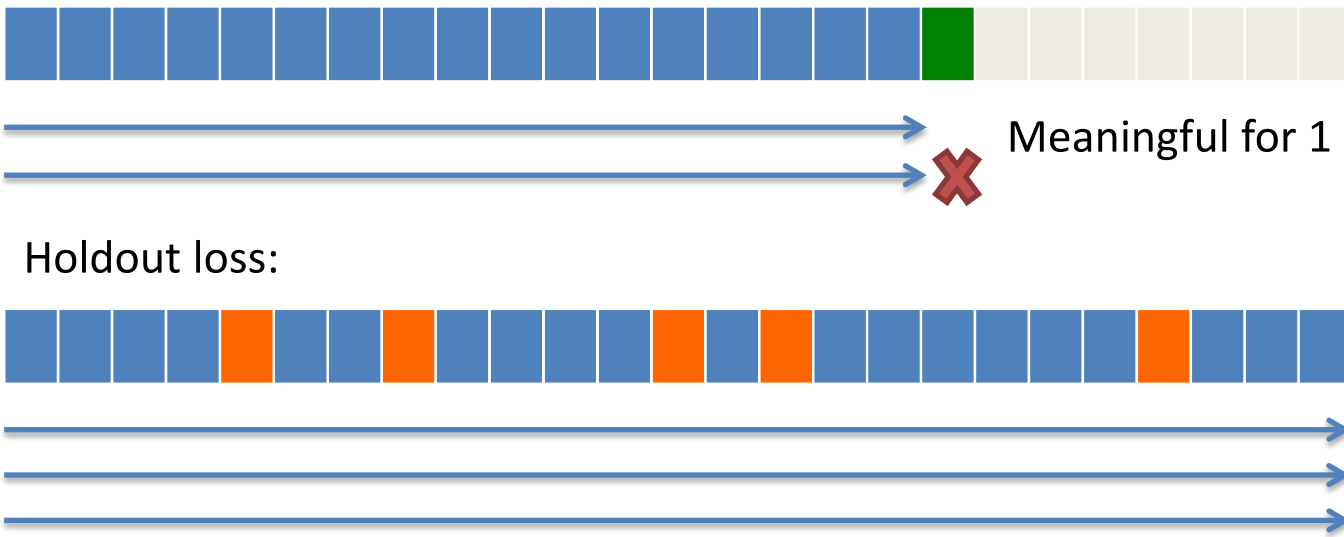
average loss	since last	example counter	example weight	current label	current predict	current features
0.666667	0.666667	2	3.0	1.0000	0.0000	5
0.558318	0.477056	5	7.0	1.0000	0.3314	5
0.475057	0.329351	8	11.0	1.0000	0.6017	5
0.239335	0.023256	17	23.0	1.0000	0.9784	5
0.125118	0.000023	33	44.0	0.0000	0.0001	5
0.063278	0.000000	65	87.0	1.0000	1.0000	5

Progressive validation loss

Will not work for multi-pass learning

NEW Holdout validation #2

Progressive validation loss:



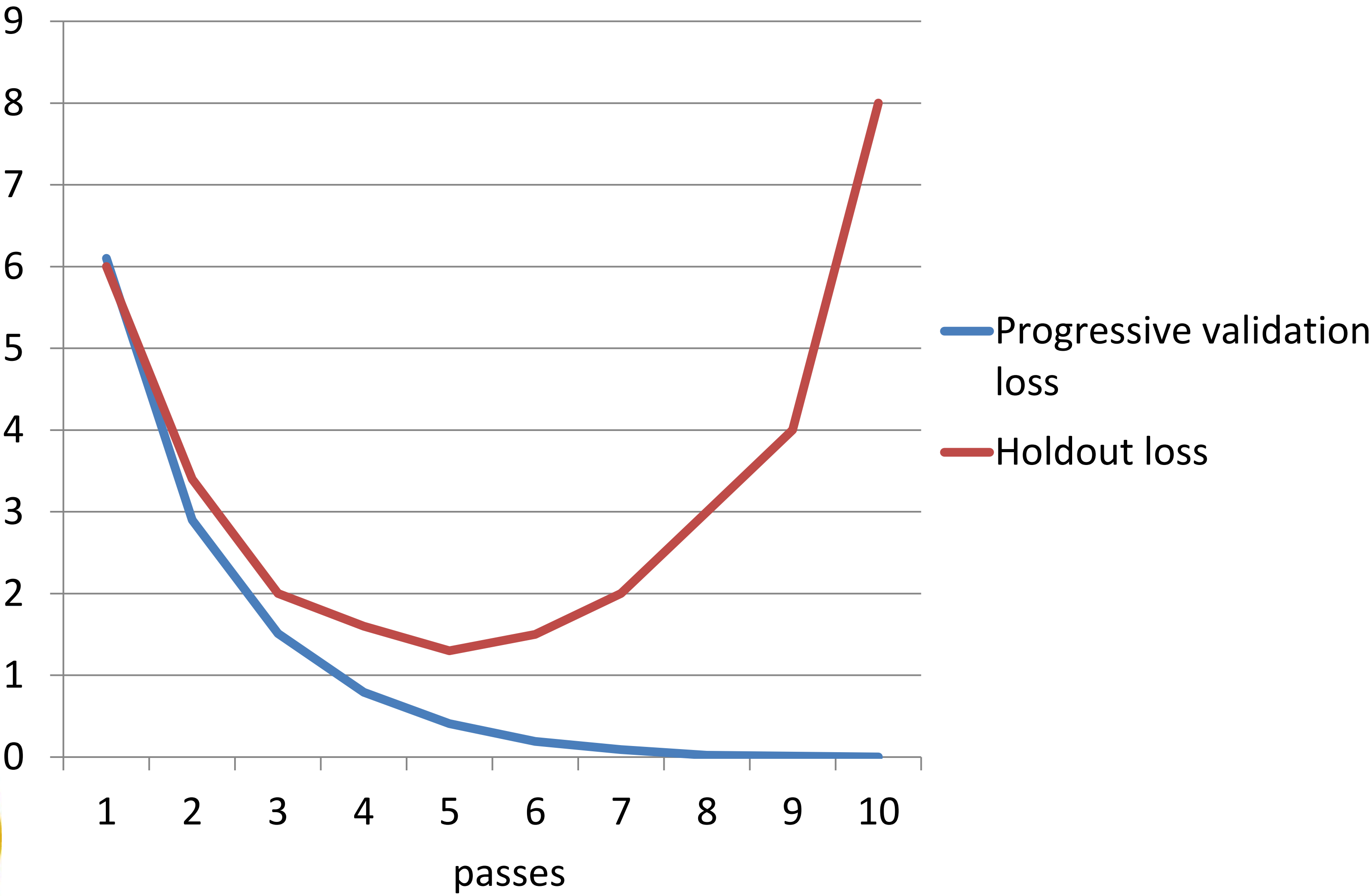
Meaningful for 1 pass only

Holdout loss:

Now: default behavior for multipass!

Holdout validation

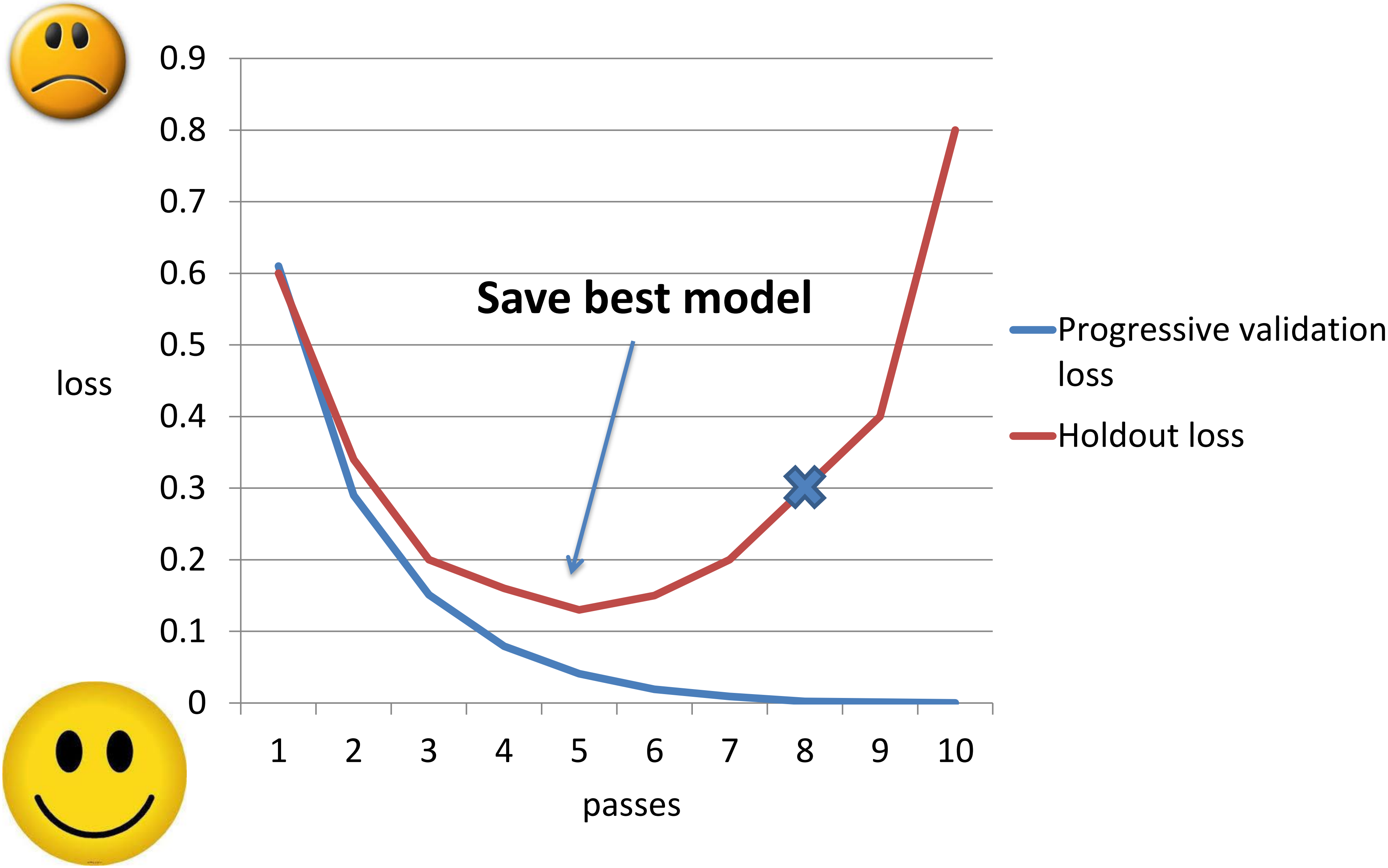
#2



NEW

--early_terminate

#3



Usage

vw ……. –f model –passes 10

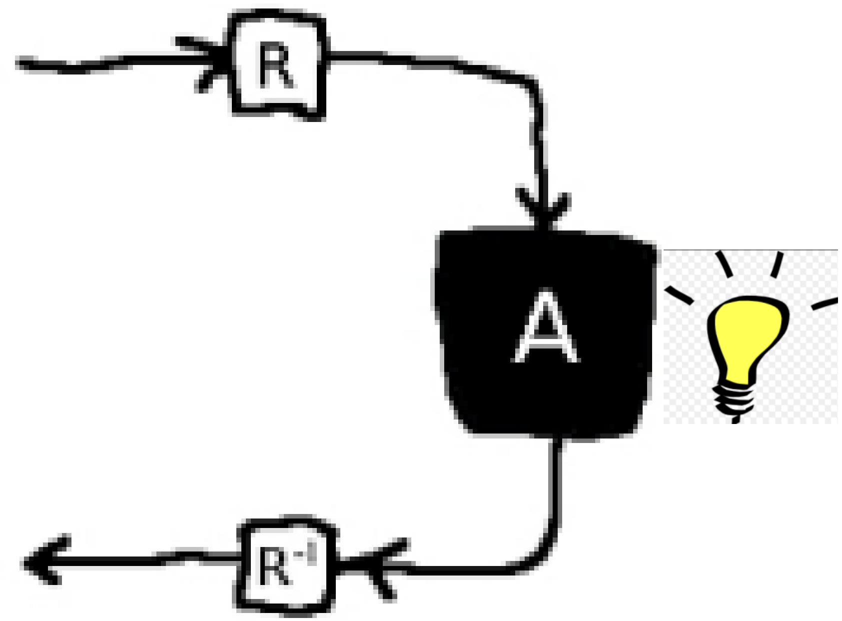
(by default)

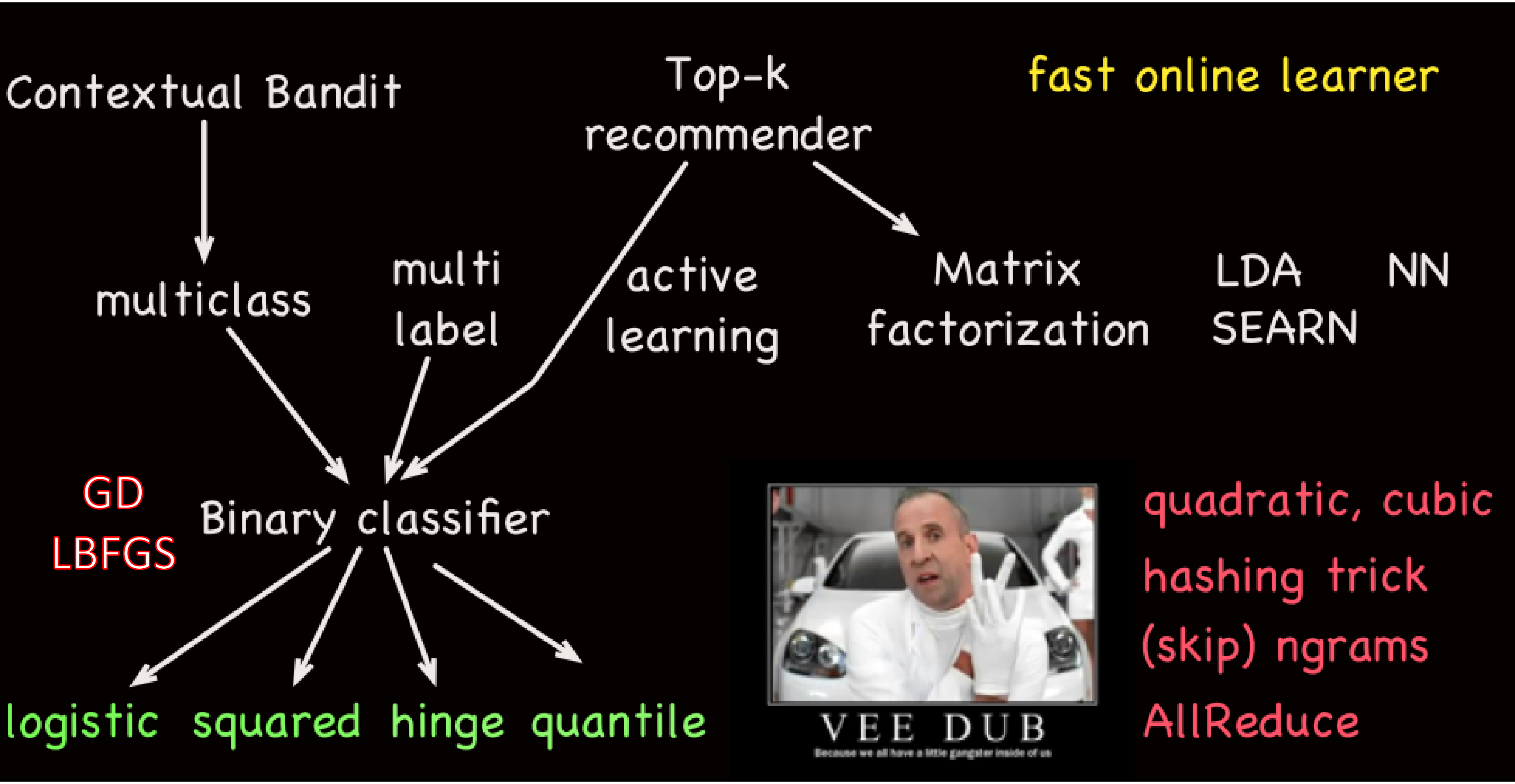
Holdout 1 in every 10 examples, print out validation loss after the 1st pass (with an ‘h’ in the end), early terminate if validation loss does not decrease for 3 passes

vw ……. –f model –passes 10 –holdout_period 5 –early_terminate 2

Vw ……. –f model –passes 10 –holdout_off

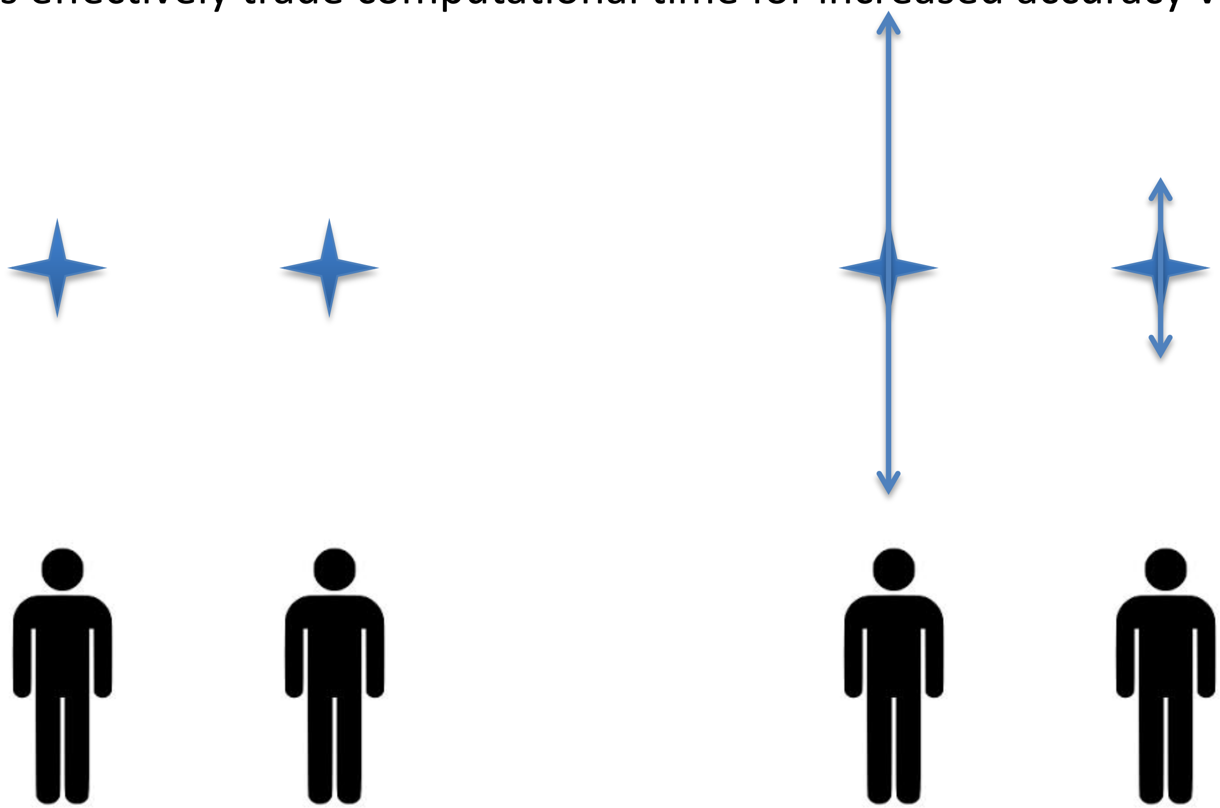
Reductions




Contextual Bandit
Top-k
recommender
fast online learner
multiclass
multi
label
active
learning
Matrix
factorization
LDA
SEARN
NN
GD
LBFGS
Binary classifier
quadratic, cubic
hashing trick
(skip) ngrams
AllReduce
logistic
squared
hinge
quantile
VEE DUB
Because we all have a little gangster inside of us

--bs: boostrapping

efficiently provides some understanding of prediction variations
sometimes effectively trade computational time for increased accuracy via ensembling



--bs: algorithm

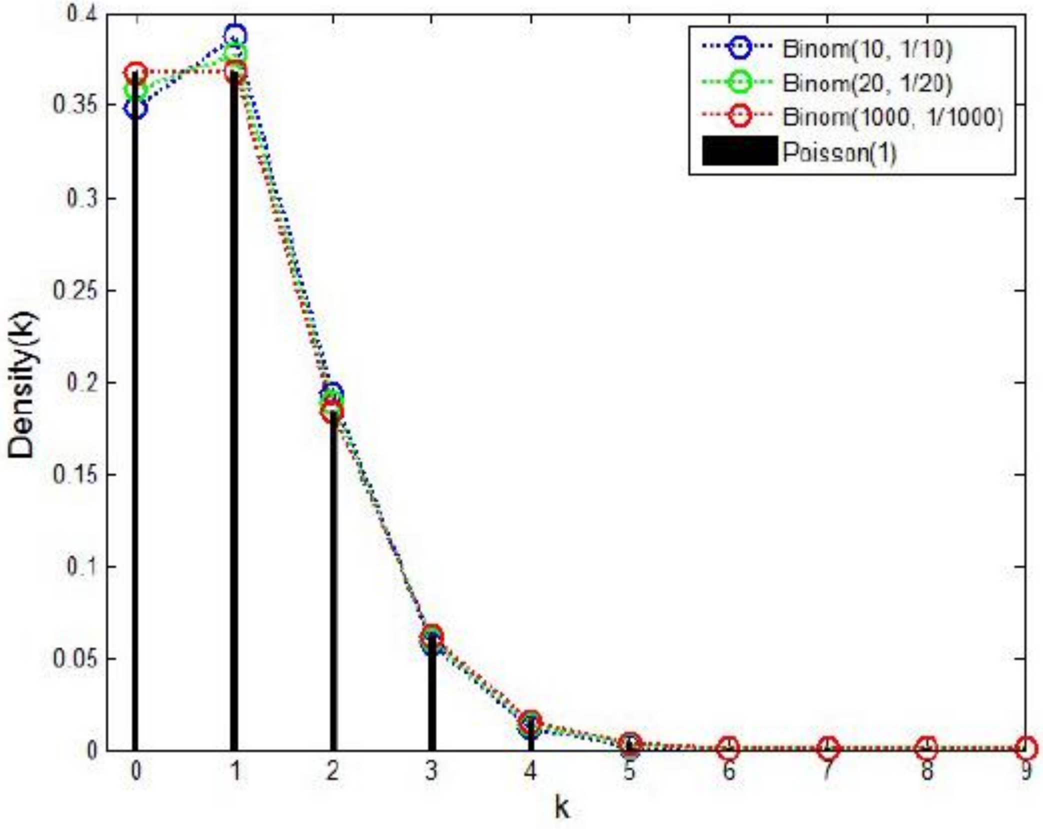
```
Input: example E with importance weight W,
       user-specified number of bootstrapping rounds N
```

```
Training:
for i = 1..N
do
  Z ~ Poisson(1) * W
  learn(E, Z, i)
done
```

```
Prediction:
for i = 1..N
do
  p_i = predict(E, i)
done
return majority(p) // or mean(p)
```

--bs: why Poisson?

Binom(n,1/n) -> Possion(n* (1/n))



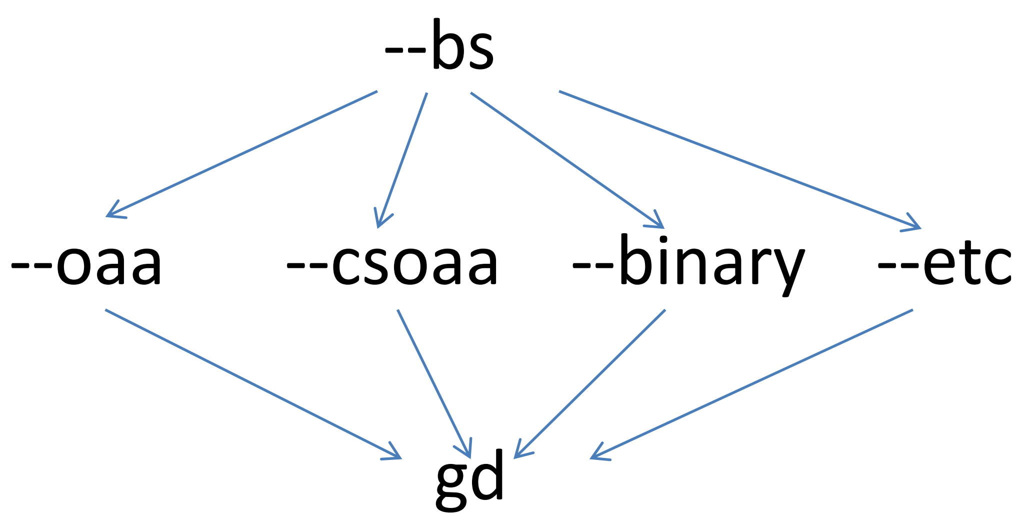
Nikunj Oza and Stuart Russell (2001)

NEW

--bs: bootstrapping

#4

Implemented as Reduction, take advantage of what is inside VW

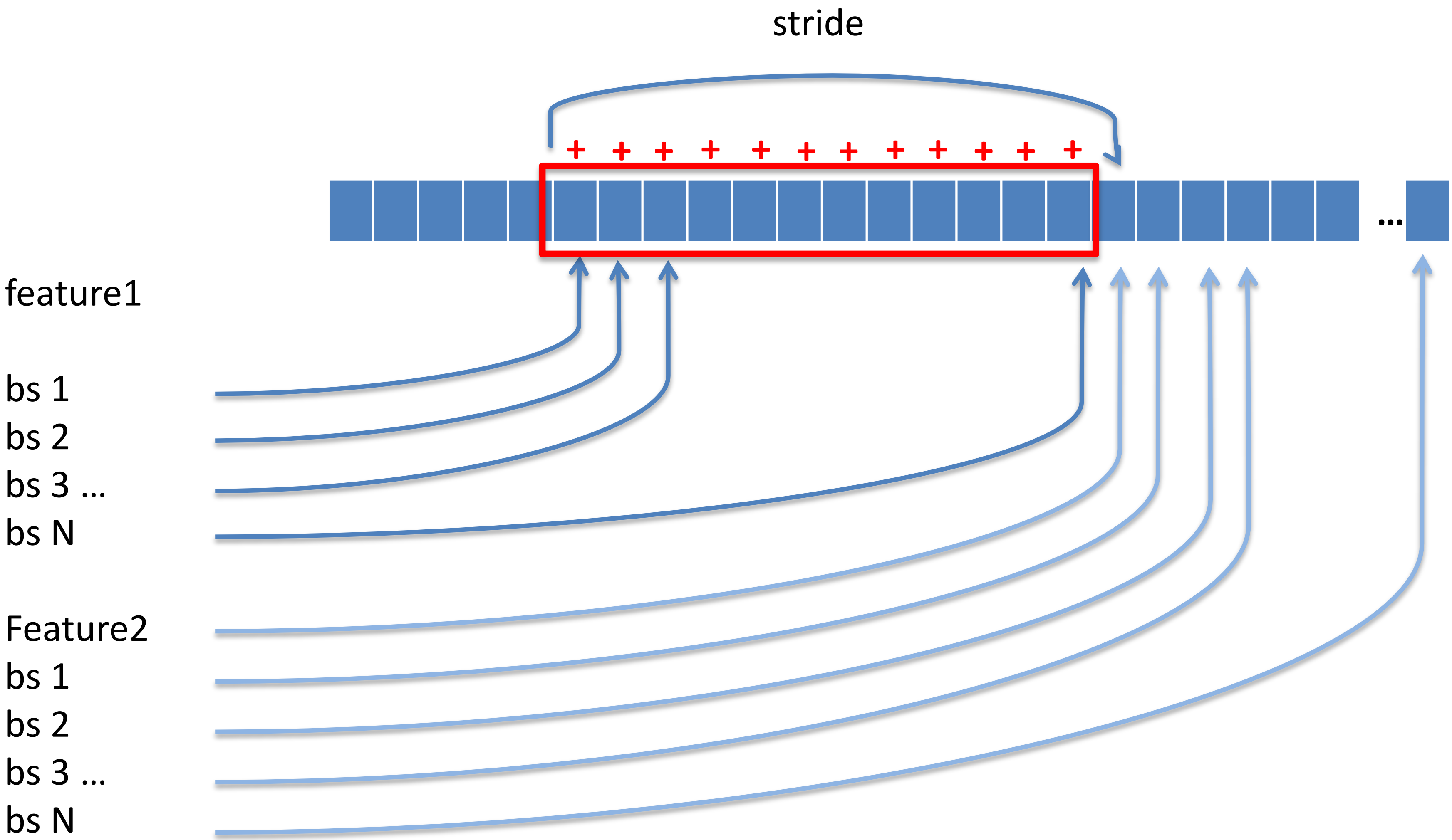


NEW

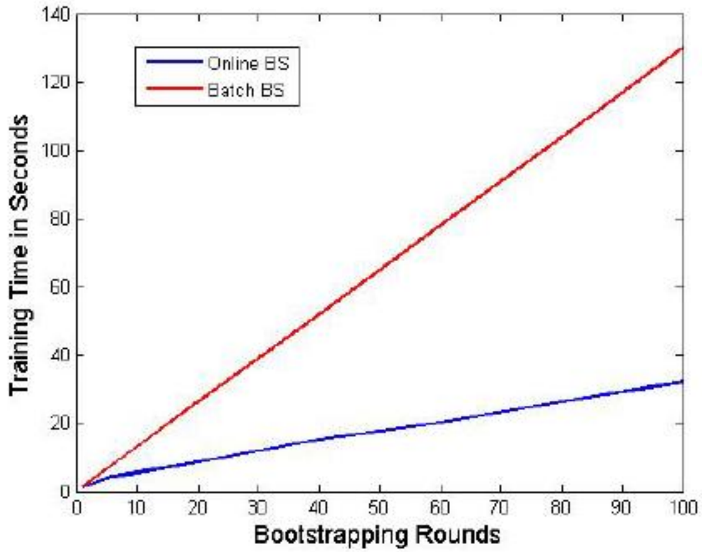
--bs: boostrapping

#4

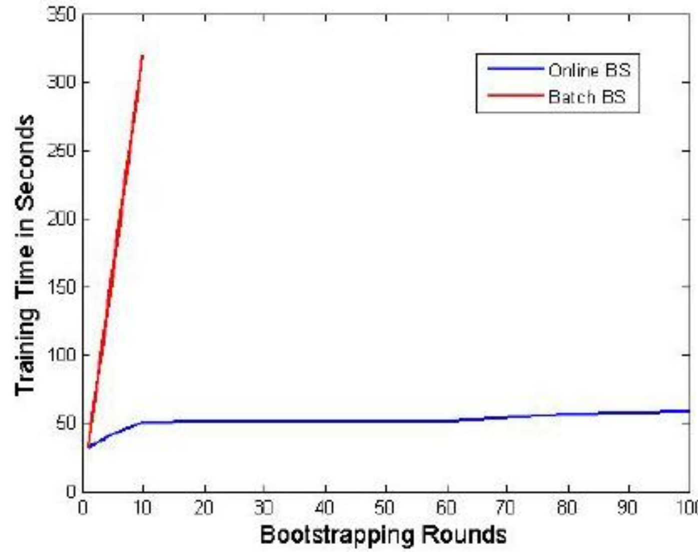
Keep memory access local and maximize cache hits



Running time



75K Dataset

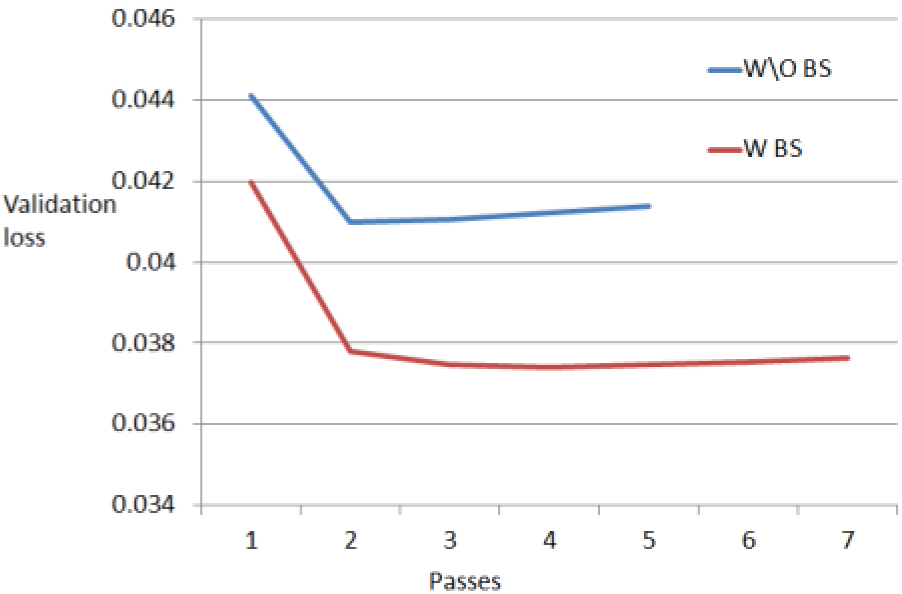


RCV1 Training

75k dataset: 74746 examples, 3000 features, 20 passes
RCV1 Training: 781265 examples, 80 features, single pass
Note RCV1 training needs heavy example parsing

--bs:bootstrapping #4

Better performance via model averaging



Method	Error Rate
Base Learner (BL)	6.01%
BL + Online BS (N=20)	5.37%
Tuned Learner (TL)	4.64%
TL + Online BS (N=4)	4.58%

RCV1 Testing data: 23149 examples

TL: Single pass learning with options -b 23 -l 0.25 --ngram 2 --skips 4

NEW

--bs: usage

#4

vw …… --bs 100 --bs_method mean (default)

vw …… --bs 100 --bs_method vote

vw …… --bs 100 --p predictions.txt

Predictions.txt
1.00 0.94 1.12
1.00 0.85 1.20
…..

NEW

--top k

#5

Choose the top k of any set of base instances
Recommendation example:

Training:
user1-movie1’ 5| features……
user1-movie2’ 3| features……
user1-movie3’ 4| features……
user1-movie4’ 1| features……
user2-movie1’ 4| features……
user2-movie2’ 4| features……
user2-movie3’ 3| features……
user2-movie4’ 2| features……

--top k

#5

Testing:
newuser1-movie1’| features……
newuser1-movie2’| features……
newuser1-movie3’| features……
newuser1-movie4’| features……

newuser2-movie1’| features……
newuser2-movie2’| features……
newuser2-movie3’| features……
newuser2-movie4’| features……

Top k recommendation for each set (separated by a newline in VW)
newuser1-movie3
newuser1-movie2

--top k

#5

Implemented as reduction, easily fits online setting

For each example E

  if E is not newline

    p <- predict(E)

    push(E,p) to a minimum priority queue with maximum size of k

  else (finished processing a set)

    print out information in pq to prediction file and clear pq

--top k usage #5

vw --d testdata -i model -t --top 2 -p predictions.txt

predictions.txt:

newuser1-movie3 3.7
newuser1-movie2 4.1

newuser2-movie4 2.7
newuser2-movie1 3.1

Summary of Contributions

NEW	--invert_hash	#1
NEW	--holdout_period	#2
NEW	--early_terminate	#3
NEW	--bs	#4
NEW	--top k	#5
NEW	-q a:, -q ::, -q :a	
NEW	--feature_mask	

Thanks!

- zqin001@cs.ucr.edu
- Poster on online bootstrapping
- Vowpal Wabbit Yahoo mailing list