Amazon.com - Employee Access Challenge

Finished

Wednesday, May 29, 2013 $5,000 • 1,691 teams Wednesday, July 31, 2013

« Prev Topic

Beating the benchmark

Next Topic »

Start Watching View all posts

< 1 2

@Lukasz: given our positions on the leaderboard, I should be asking you questions, don't you think?

Foxtrot

Posts **123**
Thanks **277**
Joined **28 Dec '11**
Email User

And now for something completely different: here's a follow-up to the original post. It's about non-linear learning with Vowpal Wabbit. With it you can up the score a little.

http://fastml.com/go-non-linear-with-vowpal-wabbit/

Thanked by **Benoit Plante** , **Martin Beyer** and **Triskelion**

#16 / Posted 4 months ago

Reply / **Quote** / **Thank** / **Flag** / **Email User**

Martin Beyer

Rank **82nd**
Posts **13**
Thanks **3**
Joined **21 Sep '12**
Email User

Here, in the attachment, is the script to generate the input for vowpal wabbit to get a 0.90624 private score, after the deadline. My best vw score before the deadline was 0.903.

The key points are:
- combined features as in Miroslaw Horbal's logistic regression code, created explicitly instead of relying on -q or --cubic
- cutoffs as explained by Nick Kridler.
- semi-supervised learning: I use the combined train- and test-data for the cutoffs
- extra features for combinations of values occurring at least twice with label 0 and never with 1 in the training set
- these extra features, called z for zero in the script and the generated data files, are represented as one boolean vw feature per column-combination, i.e. the value combinations mentioned above are lumped together. Actually, I probably never would have thought of that. It was a bug, but when I corrected it, the results got worse. The first line where they appear is 56, which I used to get via head - n 56 train.vw | tail -n 1 > train56.vw

The script still contains a few parameters for testing/playing, including some useless ones.

The generated data is then used like this:
vw -d train.vw --nn 1 -b26 -c -k -f modelb --loss_function logistic --passes 11 -l 0.41 --decay_learning_rate 0.53 --l1 5e-9

vw -t -d test.vw --nn 1 -b26 -c -k -i modelb -p predvwb.txt --loss_function logistic

If someone is still experimenting, eg. for the paper, and can use this for blending (quite improbable given the score), feel free to do so. I have to stop now.

Since I don't want to have the last word in other now inactive threads, I insert some off-topic remarks.

I used a neural net for learning and meta learning simultaneously: I fed it the data plus lots of predictions (stacked, from 10 fold cv) from logistic regression models that I got from a genetic algorithm. The results were dismal (0.900 public, 0.895 private), which doesn't prove anything.

How I changed Miroslaw's code: A nested logarithmic hyper parameter search with an epsilon criterion didn't help much.
cut -d , -f 2 submission.csv | sort | uniq | wc counts the number of distinct predictions in a submission. I got some improvement simply by increasing the number of digits in writing it.

Back to vw. Using combinations of only up to 3 features and 2 cutoffs, the generated input is small enough that I can show a line here:

-1 0.8 |d20 9} |d90 9} |d21 NZ |d91 NZ |d22 l |d92 l |d23 X |d93 X |d24 bu |d94 bu |d25 K |d95 K |d26 ss |d96 ss |d27 d |d97 d |e201 NZ?9} |e901 ? |e202 l?9} |e902 l?9} |e203 X?9} |e903 X?9} |e204 bu?9} |e904 ? |e205 K?9} |e905 K?9} |e206 ? |e906 ? |e207 d?9} |e907 d?9} |e212 l?NZ |e912 l?NZ |e213 X?NZ |e913 X?NZ |e214 bu?NZ |e914 bu?NZ |e215 K?NZ |e915 ? |e216 ss?NZ |e916 ? |e217 d?NZ |e917 ? |e223 l?X |e923 l?X |e224 l?bu |e924 l?bu |e225 l?K |e925 l?K |e226 l?ss |e926 ? |e227 d?l |e927 d?l |e234 X?bu |e934 X?bu |e235 X?K |e935 X?K |e236 X?ss |e936 ? |e237 d?X |e937 d?X |z45 ? |e245 K?bu |e945 K?bu |e246 bu?ss |e946 ? |z47 ? |e247 d?bu |e947 d?bu |e256 K?ss |e956 ? |e257 d?K |e957 d?K |e267 d?ss |e967 ? |f2012 l?NZ?9} |f9012 ? |f2013 X?NZ?9} |f9013 ? |f2014 bu?NZ?9} |f9014 ? |f2015 ? |f9015 ?

|f2016 ? |f9016 ? |f2017 ? |f9017 ? |f2023 l?X?9} |f9023 l?X?9} |f2024 l?bu?9} |f9024 ? |f2025 l?K?9} |f9025 l?K?9} |f2026 ? |f9026 ? |f2027 d?l?9} |f9027 d?l?9} |f2034 X?bu?9} |f9034 ? |f2035 X?K?9} |f9035 ? |f2036 ? |f9036 ? |f2037 d?X?9} |f9037 d?X?9} |f2045 ? |f9045 ? |f2046 ? |f9046 ? |f2047 ? |f9047 ? |f2056 ? |f9056 ? |f2057 d?K?9} |f9057 d?K?9} |f2067 ? |f9067 ? |f2123 l?X?NZ |f9123 l?X?NZ |f2124 l?bu?NZ |f9124 l?bu?NZ |f2125 l?K?NZ |f9125 ? |f2126 l?ss?NZ |f9126 ? |f2127 d?l?NZ |f9127 ? |f2134 X?bu?NZ |f9134 X?bu?NZ |f2135 X?K?NZ |f9135 ? |f2136 X?ss?

NZ |f9136 ? |f2137 d?X?NZ |f9137 ? |f2145 K?bu?NZ |f9145 ? |f2146 bu?ss?NZ |f9146 ? |f2147 d?bu?NZ |f9147 ? |f2156 K?ss?NZ |f9156 ? |f2157 d?K?NZ |f9157 ? |f2167 d?ss?NZ |f9167 ? |f2234 l?X?bu |f9234 l?X?bu |f2235 l?X?K |f9235 l?X?K |f2236 l?X?ss |f9236 ? |f2237 d?l?X |f9237 d?l?X |z245 ? |f2245 l?K?bu |f9245 l?K?bu |f2246 l?bu?ss |f9246 ? |z247 ? |f2247 d?l?bu |f9247 d?l?bu |f2256 l?K?ss |f9256 ? |f2257 d?l?K |f9257 d?l?K |f2267 d?l?ss |f9267 ? |f2345 X?K?bu |f9345 ? |f2346 X?bu?ss |f9346 ? |f2347 d?X?bu |f9347 ? |f2356 X?K?ss |f9356 ? |f2357 d?X?K |f9357 d?X?K |f2367 d?X?ss |f9367 ? |f2456 K?bu?ss |f9456 ? |z457 ? |f2457 d?K?bu |f9457 d?K?bu |f2467 d?bu?ss |f9467 ? |f2567 d?K?ss |f9567 ?

0.8 is the weight of the example. The script contains a parameter to set the weight for the lines containing z features. I found 0.8 empirically and can only speculate about why 1 is worse. Perhaps because not all lines with label 0 (-1 for vw with the logistic loss function) have a z feature, and because of overfitting.

The namespaces begin with d, e, f for combinations of 1, 2, 3 features. Using different letters is convenient for experimenting with -q, --cubic or --ignore. I's all based on Zygmunt's/Foxtrot's starter code. The last 2 namespaces are "|f2567 d?K?ss |f9567 ?": f for 3 columns, 2 and 9 are the cutoffs, 567 are the columns. d?K?ss and ? are vw feature names, with value 1. d?K?ss encodes the categories in the three columns. The script contains hree ways to do that, with this one being both short and usable for combinations of more than 4 columns. ? is used, without ambiguity, for several purposes: as a separator when concatenating the category encodings, as "other"-category in the cutoffs, and as feature name for the additional zero features.

I'd like to thank Zygmunt / Foxtrot, and everybody who contributed in this thread.

1 Attachment —

vwi-418.py (11.41 KB)

#17 / Posted 2 months ago — Reply / Quote / Thank / Flag / Email User

< 1 2