Vowpal Wabbit

http://hunch.net/~vw/

git clone
git://github.com/JohnLangford/vowpal_wabbit.git
1. **Online** by default. A good solution to an ML problem is always an hour or less away.
1. **Online** by default. A good solution to an ML problem is always an hour or less away.

2. **Hashing.** Raw text is fine. A valid input is:
   
   1  | The dog ate my homework
1. **Online** by default. A good solution to an ML problem is always an hour or less away.

2. **Hashing**. Raw text is fine. A valid input is: 1 | The dog ate my homework

3. **Allreduce**. Terascale learning paper = most scalable public algorithm.
1. **Online** by default. A good solution to an ML problem is always an hour or less away.

2. **Hashing.** Raw text is fine. A valid input is:
   
   1 | The dog ate my homework

3. **Allreduce.** Terascale learning paper = most scalable public algorithm.

4. **Reductions.** Solve wide variety of problems well by reduction to simple problems.
1. **Online** by default. A good solution to an ML problem is always an hour or less away.

2. **Hashing.** Raw text is fine. A valid input is:
   
   ```
   1 | The dog ate my homework
   ```

3. **Allreduce.** Terascale learning paper = most scalable public algorithm.

4. **Reductions.** Solve wide variety of problems well by reduction to simple problems.

5. **Interactive.** Causation instead of correlation. Learn to control based on feedback.
1. **Online** by default. A good solution to an ML problem is always an hour or less away.

2. **Hashing.** Raw text is fine. A valid input is: 1 | The dog ate my homework

3. **Allreduce.** Terascale learning paper = most scalable public algorithm.

4. **Reductions.** Solve wide variety of problems well by reduction to simple problems.

5. **Interactive.** Causation instead of correlation. Learn to control based on feedback.

6. **Learn2Search.** See Hal.
1. Online by default. A good solution to an ML problem is always an hour or less away.

2. Hashing. Raw text is fine. A valid input is:
   1 | The dog ate my homework

3. Allreduce. Terascale learning paper = most scalable public algorithm.

4. Reductions. Solve wide variety of problems well by reduction to simple problems.

5. Interactive. Causation instead of correlation. Learn to control based on feedback.


7. others....
A user base becomes addictive

1. Mailing list of about 400
A user base becomes addictive

1. Mailing list of about 400
2.
vw -c rcv1.train.raw.txt -b 22 --ngram 2 --skips 4 -l 0.25 --binary provides stellar performance in 12 seconds.
Learning Reductions

The core idea: reduce complex problem A to simpler problem B then use solution on B to get solution on A.

Problems:

1. How do you make it efficient enough?
2. How do you make it natural to program?
void learn(learner& base, example* ec) {
    base.learn(ec); // The recursive call
    if (ec->final_prediction > 0) // Thresholding
        ec->final_prediction = 1;
    else
        ec->final_prediction = -1;
    label_data* ld = (label_data*)ec->ld; // New loss
    if (ld->label == ec->final_prediction)
        ec->loss = 0.;
    else
        ec->loss = 1.;
}

Structured Prediction

= joint prediction with a joint loss
Structured Prediction

= joint prediction with a joint loss

Example: Part of Speech Tagging

Pierre | Vinken | , | 61 | years | old
Proper N. | Proper N. | Comma | Number | Noun | Adj.
Structured Prediction

= joint prediction with a joint loss

Example: Part of Speech Tagging

Pierre  |  Vinken  |  ,  |  61  |  years  |  old
Proper N.  |  Proper N.  |  Comma  |  Number  |  Noun  |  Adj.

Example 2: Machine Translation

縄文 土偶 : 仮面 の 女神 、 国宝 指定
|  |  |  |  |
Jomon figurines : goddess of Kamen , designated a national treasure
How can you best do structured prediction?

We care about:

1. Programming complexity.
How can you best do structured prediction?

We care about:

1. **Programming complexity.** Most structured predictions are not addressed with structured learning algorithms, because it is too complex to do so.
How can you best do structured prediction?

We care about:

1. Programming complexity. Most structured predictions are not addressed with structured learning algorithms, because it is too complex to do so.

2. Prediction accuracy. It had better work well.
How can you best do structured prediction?

We care about:

1. **Programming complexity.** Most structured predictions are not addressed with structured learning algorithms, because it is too complex to do so.
2. **Prediction accuracy.** It had better work well.
3. **Train speed.** Debug/development productivity + maximum data input.
How can you best do structured prediction?

We care about:

1. Programming complexity. Most structured predictions are not addressed with structured learning algorithms, because it is too complex to do so.
2. Prediction accuracy. It had better work well.
3. Train speed. Debug/development productivity + maximum data input.
4. Test speed. Application efficiency
A program complexity comparison

![Graph showing lines of code for POS](image)

- CRFSGD
- CRF++
- S-SVM
- Search
Part of speech tagging (tuned hps)

Accuracy (per tag)

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>VW Search</td>
<td>95.7</td>
</tr>
<tr>
<td>VW Search (own fts)</td>
<td>96.6</td>
</tr>
<tr>
<td>VW Classification</td>
<td>95.0</td>
</tr>
<tr>
<td>CRF SGD</td>
<td>95.8</td>
</tr>
<tr>
<td>CRF++</td>
<td>96.1</td>
</tr>
<tr>
<td>Str. Perceptron</td>
<td>96.1</td>
</tr>
<tr>
<td>Structured SVM</td>
<td>95.3</td>
</tr>
<tr>
<td>Str. SVM (DEMI-DCD)</td>
<td>90.7</td>
</tr>
</tbody>
</table>

Training Time (minutes)

- 0.88
- 0.90
- 0.92
- 0.94
- 0.96
- 0.98
An example

vw -b 24 -d wsj.train.vw -c --search_task sequence --search 45 --search_alpha 1e-8 --search_neighbor_features -1:w,1:w --affix -1w,+1w

Good performance in about 2.5 minutes.