Tuning as Linear Regression

Marzieh Bazrafshan, Tagyoung Chung, and Daniel Gildea

University of Rochester
Department of Computer Science

June 5th, 2012
Parameter tuning: finding feature weights for a linear translation model

Minimum Error Rate Training (MERT): Incorporate evaluation criteria (often BLEU)

MERT does not scale

MERT only maximizes the score for the 1-best hypothesis
Introduction

- Alternative to MERT: Margin Infused Relaxed Algorithm (MIRA)
- MIRA scales but has a harder optimization problem
- MIRA needs modifications to decoder
Introduction

- Hopkins and May 2011: Pairwise Ranking Optimization (PRO)
  - Tuning by ranking (“correctly ordered” and “incorrectly ordered”)
  - Maximum entropy based classifier

- This paper: Using linear regression instead
  - Linear regression simpler than maximum entropy classifier
Tuning as Ranking (Hopkins and May, 2011)

- PRO casts tuning as ranking pairs of sentences.
- Goal: Use the gold scoring function to rank pairs.
- Uses k-best candidate translations
- Binary classification to learn the rankings.
Tuning as Ranking (Hopkins and May, 2011)

- **Goal:**
  \[ g(e_1) > g(e_2) \iff h(e_1) > h(e_2) \]
  
  \( g(e) \): gold score of \( e \) (BLEU)
  
  \( h(e) \): model score \( w^T x(e) \)

- **Substituting** \( h(e) \) in the above biconditional:
  \[ g(e_1) - g(e_2) > 0 \iff w^T (x(e_1) - x(e_2)) > 0 \]

- **Binary classification to learn** \( w \):
  - Data points: \( x(e_1) - x(e_2) \)
  - Targets: \( \text{sign}(g(e_1) - g(e_2)) \)
  - \( g(e) \): BLEU+1
  - MegaM classifier
Tuning as Linear Regression

- Idea: Changing the maximum entropy classification of PRO to simple linear regression:

\[ g(e_1) - g(e_2) = w^T(x(e_1) - x(e_2)) \]

- Use the least squares method:

\[ w = (X^T X)^{-1} X^T g \]

- Adding \( L_2 \) regularization with parameter \( \lambda \):

\[ w = (X^T X + \lambda I)^{-1} X^T g \]

- \( X \): matrix of the data points
- \( g \): vector of the targets
- \( w \): weight vector that is learned
Tuning as Linear Regression

Preparing $\mathbf{X}$ and $\mathbf{g}$ (following Hopkins and May, 2011):

- For each sentence:
  1. Generate $k$-best list
  2. Sample $n$ pairs
  3. Calculate gold scores $g$
  4. Keep pairs with difference higher than $t$
  5. Pick $s$ pairs with highest differences
  6. For each pair, make two data points
     $$(h(e_1) - h(e_2), g(e_1) - g(e_2))$$ and
     $$(h(e_2) - h(e_1), g(e_2) - g(e_1))$$

- $\mathbf{X}$: $h(e_1) - h(e_2)$
- $\mathbf{g}$: $g(e_1) - g(e_2)$
Model Score: $w^T x$

Gold score: $g$

$g(e_1) - g(e_2) > 0 \iff w^T (x(e_1) - x(e_2)) > 0$

- Implies monotonicity
- Potentially a complex function
The sizes of the intervals must correlate
  - Can be too strong of an assumption
  - Can help reduce overfitting
Experiments - Setup

- Training Data: Chinese-English corpus from LDC with parsed English side (250K sentence pairs, 6.3M words)
- Development Set: 392 pairs
- Test Set: 428 pairs
- 4 reference translations
- Translation Rules: GHKM rules
Features: 9 features including globally normalized count of rules, lexical weighting, and length penalty, etc.

Decoder: syntax-based, Earley parsing with cube pruning

Parameters:
- 1500-best lists (k)
- 5000 sample pairs (n)
- 0.05 difference threshold (t)
- 50 kept sample pairs (s)
Results for 8 experiments with random initial weight vectors:

<table>
<thead>
<tr>
<th></th>
<th>Average of max BLEU</th>
<th>Max BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>dev</td>
<td>test</td>
</tr>
<tr>
<td>Regression</td>
<td>27.7 (0.91)</td>
<td>26.4 (0.82)</td>
</tr>
<tr>
<td>PRO</td>
<td>26.9 (1.05)</td>
<td>25.6 (0.84)</td>
</tr>
</tbody>
</table>

- Numbers in the parentheses are the standard deviations
- Each experiments run for 25 iterations
Average of eight runs of linear regression and PRO:
Experiments - Results

- Linear regression reaches the maximum faster than PRO (avg. 14 vs. 20 iterations)
- Both training times negligible compared to decoding
- Regularization: smaller variance but not much effective for overfitting (few features)
Conclusion

- Linear regression comparable to PRO and on average faster
- Binary classification with signs vs. regression with the magnitude of the differences
- Linear regression simpler than maximum entropy binary classification
Conclusion

- 1-best sentence by MERT vs. k-best ranking by PRO and linear regression
- Scaling to large number of features
- Linear regression similar to MIRA in using the magnitudes
- Both Simpler to implement than MIRA
Questions?