Speech and Language Processing

Lecture 6
Weighted finite state transducer (WFST) and speech decoding

Information and Communications Engineering Course
Takahiro Shinozaki
Lecture Plan (Shinozaki’s part)

I gives the first 6 lectures about speech recognition. Through these lectures, the backbone of the latest speech recognition techniques is explained.

1. 10/19 (remote)  
   Speech recognition based on GMM, HMM, and N-gram

2. 10/26 (remote)  
   Maximum likelihood estimation and EM algorithm

3. 11/5 (@TAIST)  
   Bayesian network and Bayesian inference

4. 11/5 (@TAIST)  
   Variational inference and sampling

5. 11/6 (@TAIST)  
   Neural network based acoustic and language models

6. 11/6 (@TAIST)  
   Weighted finite state transducer (WFST) and speech decoding
Today’s Topic

• Answers for the previous exercises
• Weighted finite state transducer (WFST)
• Speech decoding
Answers for the Previous Exercises
Exercise 5.1

• Let \( h \) be a softmax function having inputs \( z_1, z_2, \ldots, z_N \).

\[
h(z_i) = \frac{\exp(z_i)}{\sum_j \exp(z_j)}
\]

• Prove that \( \sum_{i=1}^{N} h(z_i) = 1.0 \)
Exercise 5.2

When \( h(y) \) and \( y(x) \) are given as follows, obtain \( \frac{\partial h}{\partial x} \)

\[
h(y) = \frac{1}{1 + \exp(-y)}
\]

\( y = ax + b \)
Weighted finite state transducer (WFST)
Weighted Finite State Acceptor (WFSA)

- Defined by a set of nodes, arcs, arc labels, and arc weights
- An extension of finite state automaton
- Accepts a sequence of symbols (string) assigning a weight to the sequence
## Variations of Weight Types

<table>
<thead>
<tr>
<th>Name</th>
<th>Set</th>
<th>Plus</th>
<th>Times</th>
<th>Zero</th>
<th>One</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td>{0,1}</td>
<td>\lor</td>
<td>\land</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Real</td>
<td>{0, \infty}</td>
<td>+</td>
<td>*</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Log</td>
<td>{-\infty, \infty}</td>
<td>-\log(e^{-x}+e^{-y})</td>
<td>+</td>
<td>\infty</td>
<td>0</td>
</tr>
<tr>
<td>Tropical</td>
<td>{-\infty, \infty}</td>
<td>min</td>
<td>+</td>
<td>\infty</td>
<td>0</td>
</tr>
</tbody>
</table>
Weight of Path (Tropical Weight)

Weight of accepting “a,b” is:
\[ w_1 \otimes w_2 = w_1 + w_2 \]

Weight of accepting “a” is:
\[ w_1 \oplus w_2 = \min\{w_1, w_2\} \]
Example

Initial state

Final state

Tropical Weight

\(<a, a, a, b> (1.4)\\n\(<b, b, b> (0.4)\\n
Weighted Finite State Transducer

- Defined by a set of nodes, arcs, input/output arc labels, and arc weights
- An extension of finite state acceptor
- Transduces an input string to an output string, assigning a weight
Example

Initial state

Final state

Tropical Weight

\[ <a, a, a, b> \rightarrow <A, B, B, B> (1.4) \]
\[ <b, b, b> \rightarrow <B, C, B> (0.4) \]
Null ($\varepsilon$) Symbol

- Input symbol is $\varepsilon$: No input
- Output symbol is $\varepsilon$: No output

\[
\begin{align*}
\langle B, A \rangle & \rightarrow \langle a, b, a \rangle (2.5) \\
\langle C, A \rangle & \rightarrow \langle a \rangle (1.2)
\end{align*}
\]
Invert of WFST

• Swap input symbol and output symbol
Equivalence of WFST

- Two transducers are equivalent if for each input string, they produce the same output strings with the same weight.

\[ a:A/0.5 \quad \text{Equivalent} \quad \varepsilon:A/0.3 \]

\[ a:\varepsilon/0.2 \quad \text{Tropical Weight} \]
Determinization of WFST

- Makes equivalent WFST so that no state has two transitions with the same input label

Tropical Weight

WFST (A)

WFST (B)
Exercise 6.1

• Check that WFST (A) and (B) in previous slide are equivalent by enumerating all possible input strings
Composition of WFSTs

- WFST1 transduces string x to string y with weight \( w_1 \)
- WFST2 transduces string y to string z with weight \( w_2 \)
- Composition of WFST1 and WFST2 (\( WFST1 \cdot WFST2 \)) makes WFST that transduces x to z with weight \( w_1 \otimes w_2 \)

WFST (A): \( \varepsilon: \varepsilon/0.6 \)

a:A/0.4 → b:B/1.0

WFST (B): B:い/0.2

A:あ/0.8

Composition

WFST (C): b:い/1.8

Tropical Weight
Exercise 6.2

- Check that WFST (C) in previous slide is equivalent to WFST (A) • WFST(B) by enumerating all possible input strings
Probability Distribution and WFST

<table>
<thead>
<tr>
<th></th>
<th>A=a1</th>
<th>A=a2</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(A)</td>
<td>0.4</td>
<td>0.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>B=b1</th>
<th>B=b2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A=a1</td>
<td>0.2</td>
<td>0.8</td>
</tr>
<tr>
<td>A=a2</td>
<td>0.3</td>
<td>0.7</td>
</tr>
</tbody>
</table>

Real Weight

A

B

a1:a1/0.4

a2:a2/0.6

a1:b1/0.2

a1:b2/0.8

a2:b1/0.3

a2:b2/0.7
Composition and Joint Probability

WFST A

\[ a1:a1/0.4 \]
\[ a2:a2/0.6 \]

WFSTB

\[ a1:b1/0.2 \]
\[ a1:b2/0.8 \]
\[ a2:b1/0.3 \]
\[ a2:b2/0.7 \]

\[ P(A,B) \]
\[ \propto P(B|A=a) \]

Inv(B) \cdot Inv(A)

\[ P(A,B) \]
\[ \propto P(A|B=b) \]

Real Weight

\[ a1:b1/0.08 \]
\[ a1:b2/0.32 \]
\[ a2:b1/0.18 \]
\[ a2:b2/0.42 \]
\[ b1:a1/0.8 \]
\[ b2:a1/0.32 \]
\[ b1:a2/0.18 \]
\[ b2:a2/0.42 \]
Speech decoding
WFST Representation of N-gram

Real Weight

Bi-gram WFST

WFST 24Bi-gram WFST wN:wN/P(wN|Start)

End:End/P(End|w2)

w1:w1/P(w1|Start)

Start:Start/1.0
WFST Representation of HMM

HMM of phone [a]

WFST

Log weight
Exercise 6.3

• Make a WFST that represents the following bi-gram

<table>
<thead>
<tr>
<th>C\W</th>
<th>today</th>
<th>is</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start</td>
<td>0.6</td>
<td>0.3</td>
<td>0.1</td>
</tr>
<tr>
<td>today</td>
<td>0.2</td>
<td>0.5</td>
<td>0.3</td>
</tr>
<tr>
<td>is</td>
<td>0.4</td>
<td>0.1</td>
<td>0.5</td>
</tr>
</tbody>
</table>

*P(Start)=1.0
WFST To Recognize Phone Sequence

Input: Sequence of HMM state names
Output: Sequence of phonemes
Obtain an output symbol sequence and its weight for an input sequence $<i_{s1}, i_{s1}, i_{s1}, i_{s2}, i_{s3}, i_{s3}, i_{s1}, i_{s2}, i_{s3}>$. Assume Real weight. When you answer, remove NULL (ε) symbol.
Construction of a Speech Recognition System

• Prepare:
  • WFST representing phone HMMs
  • WFST representing pronunciation dictionary: L
  • WFST representing word network or N-gram: G

• Compose:
  • WFST(Recognition system)
    = H ⋅ L ⋅ G
  • The WFST H⋅L⋅G has input labels indicating a HMM state name (=emission distribution) and output labels of a word

An index of an emission distribution

Word or ε

Weight

i:o/0.3
Speech Decoding based on WFST

- Starting from the initial state, $t$-th transition with non-epsilon input label $s_i$ corresponds to $t$-th time frame $x_t$ in input acoustic feature sequence.
- By computing acoustic probability and adding it to language probability, an WFSA is obtained.
Search the Best Recognition Hypothesis

• By performing minimum cost path search on the obtained WFSA, a word sequence that best matches to the input sound is obtained as the sequence of the output labels

• The popular search algorithm is Viterbi beam search
Quick Home Work

• Q5.3, Q6.4
• Due: 21:00 Tomorrow
• Submission:
  • Attach to an email: shinot@ict.e.titech.ac.jp
  • Title: TAISTQ53Q64
  • Format: Text file
  • file name: your student ID (eg. 012345789.txt)

Your name, Your ID
Q5.3
yes
Q6.4
<a,i,u>
0.1234567
Appendix
Tools for WFST

• OpenFST
  • http://www.openfst.org

• Graphviz
  • http://www.graphviz.org
Example

```
0 1 <eps> a 0.5
0 1 C c 0.3
0 2 C <eps> 0.2
1 2 B b 1.0
2 3 A a
3
```

```
<eps> 0
A 1
B 2
C 3
```

```
<eps> 0
a 1
b 2
c 3
```

```
$ fstcompile --isymbols=isym.txt --osymbols=osym.txt --keep_isymbols --keep_osymbols
wfst1.txt wfst1.fst
$ fstdraw wfst1.fst | dot -Tpdf > wfst1.pdf
$ evince wfst1.pdf &
```
Textbooks of Machine Learning

• Christopher Bishop,
  Pattern Recognition and Machine Learning,
  Springer-Verlag New York, 2006

• Kevin P. Murphy,
  Machine Learning A Probabilistic Perspective,
  The MIT Press, 2012