Tree Kernels for NLP

Apoorv Agarwal

June 23, 2010
Outline

1. What are Kernels?
   - Definition
   - Broad Classification of Kernels
   - Example: String Kernels
   - Developing your own Kernel
Outline

1. What are Kernels?
   - Definition
   - Broad Classification of Kernels
   - Example: String Kernels
   - Developing your own Kernel

2. Applications of Tree Kernels for NLP
Outline

1. What are Kernels?
   - Definition
   - Broad Classification of Kernels
   - Example: String Kernels
   - Developing your own Kernel

2. Applications of Tree Kernels for NLP

3. Reading List
1. What are Kernels?
   - Definition
   - Broad Classification of Kernels
   - Example: String Kernels
   - Developing your own Kernel

2. Applications of Tree Kernels for NLP

3. Reading List

4. Issues/Questions
(Simplistic Definition) A Kernel is a function that takes as input feature vectors or discrete objects and returns the measure of their similarity in high dimensional space.
What is a Kernel?

(Simplistic Definition) A Kernel is a function that takes as input feature vectors or discrete objects and returns the measure of their similarity in high dimensional space.

  - Tree Kernels (Collins and Duffy 2002)
  - Sub-sequence String Kernels (Lodhi et al 2002)
  - Spectrum Kernels (Leslie et al 2002)
  - ...

Broad Classification of Kernels

$D_1$  $D_2$
Broad Classification of Kernels

Feature Vector for $D_1$

Feature Vector for $D_2$
Broad Classification of Kernels

$D_1 = \text{String}_1$

$D_2 = \text{String}_2$

Feature Vector for $D_1$

Feature Vector for $D_2$

Kernel Trick
- Gaussian
- Poly
- RBF
Broad Classification of Kernels

Kernel Trick
- Gaussian
- Poly
- RBF

Feature Vector for $D_1$

Feature Vector for $D_2$

$D_1 = String_1$

$D_2 = String_2$
Broad Classification of Kernels

Feature Vector for $D_1$

Feature Vector for $D_2$

$D_1 = String_1$

$D_2 = String_2$

Kernel Trick
- Gaussian
- Poly
- RBF

Dynamic Prog.
- String Kernel
String Kernels (Lodhi et al 2002)

\[ K_n(s, t) = \sum_{u \in \sum^n} \langle \phi_u(s) \cdot \phi_u(t) \rangle \]

- \( K_n(s, t) \): Kernel Function
- \( s = D_1, t = D_2 \)
- \( \phi_u(.) \): Feature expansion function (or implicit mapping)
- \( u \): Subsequence of a string
- \( \sum \): Alphabet
- \( n \): Length of a subsequence
Example

\[ K_n(s, t) = \sum_{u \in \sum^n} \langle \phi_u(s) \cdot \phi_u(t) \rangle \]

- cat crate
Example

\[ K_n(s, t) = \sum_{u \in \Sigma^n} \langle \phi_u(s), \phi_u(t) \rangle \]
Example

- cat crate
- cat
- 2

\[ K_n(s, t) = \sum_{u \in \Sigma^n} \langle \phi_u(s) \cdot \phi_u(t) \rangle \]
$K_n(s, t) = \sum_{u \in \Sigma^n} \langle \phi_u(s) \cdot \phi_u(t) \rangle$

- cat crate
- cat
- $2$
- $\{aa, ab, ac, \ldots, zz\}$
Example

- cat crate
- cat
- $K_n(s, t) = \sum_{u \in \Sigma^n} \langle \phi_u(s) \cdot \phi_u(t) \rangle$
- 2
- \{aa, ab, ac, \ldots, zz\}
- [a-z]
\[
K_n(s, t) = \sum_{u \in \Sigma^n} \langle \phi_u(s) \cdot \phi_u(t) \rangle
\]

- cat crate
- cat
- \{aa, ab, ac, ... , zz\}
- [a-z]
- \{c-a, c-t, a-t\}
- 2
Example

- cat crate
- cat

\[
K_n(s, t) = \sum_{u \in \sum^n} \langle \phi_u(s) \cdot \phi_u(t) \rangle
\]

- 2
- \{aa, ab, ac, \ldots, zz\}
- [a-z]
- \{c-a, c-t, a-t\}
- \{c-a, c-t, c-c, c-r, c-e, a-c, \ldots\}
Continuing with the Example...

\[ K_n(s, t) = \sum_{u \in \sum^n} \langle \phi_u(s) \cdot \phi_u(t) \rangle \]

<table>
<thead>
<tr>
<th></th>
<th>a-t</th>
<th>c-a</th>
<th>c-t</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \phi_2(\text{cat}) )</td>
<td>( \lambda^2 )</td>
<td>( \lambda^2 )</td>
<td>( \lambda^3 )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>a-t</th>
<th>c-a</th>
<th>c-t</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \phi_2(\text{cat crate}) )</td>
<td>( \lambda^2 + \lambda^2 + \lambda^7 )</td>
<td>( \lambda^2 + \lambda^3 + \lambda^7 )</td>
<td>( \lambda^3 + \lambda^4 + \lambda^8 )</td>
</tr>
</tbody>
</table>
Continuing with the Example...

\[ K_n(s, t) = \sum_{u \in \sum^n} \langle \phi_u(s) \cdot \phi_u(t) \rangle \]

<table>
<thead>
<tr>
<th>.</th>
<th>a-t</th>
<th>c-a</th>
<th>c-t</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\phi_2(\text{cat}))</td>
<td>(\lambda^2)</td>
<td>(\lambda^2)</td>
<td>(\lambda^3)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>.</th>
<th>a-t</th>
<th>c-a</th>
<th>c-t</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\phi_2(\text{cat crate}))</td>
<td>(\lambda^2 + \lambda^2 + \lambda^7)</td>
<td>(\lambda^2 + \lambda^3 + \lambda^7)</td>
<td>(\lambda^3 + \lambda^4 + \lambda^8)</td>
</tr>
</tbody>
</table>

- Normalization?
Continuing with the Example...

\[ K_n(s, t) = \sum_{u \in \sum^n} \langle \phi_u(s) \cdot \phi_u(t) \rangle \]

<table>
<thead>
<tr>
<th></th>
<th>a-t</th>
<th>c-a</th>
<th>c-t</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \phi_2(\text{cat}) )</td>
<td>( \lambda^2 )</td>
<td>( \lambda^2 )</td>
<td>( \lambda^3 )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>a-t</th>
<th>c-a</th>
<th>c-t</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \phi_2(\text{cat crate}) )</td>
<td>( \lambda^2 + \lambda^2 + \lambda^7 )</td>
<td>( \lambda^2 + \lambda^3 + \lambda^7 )</td>
<td>( \lambda^3 + \lambda^4 + \lambda^8 )</td>
</tr>
</tbody>
</table>

- Normalization?
- How to use DP to calculate the Kernel?
What we’re essentially doing is...

- Defining our discrete object (String)
What we’re essentially doing is...

- Defining our discrete object (String)
- Defining implicit feature expansion (subsequences of length $n$)
What we’re essentially doing is...

- Defining our discrete object (String)
- Defining implicit feature expansion (subsequences of length $n$)
- Automatically defines the similarity measure (dot product)
What we’re essentially doing is...

- Defining our discrete object (String)
- Defining implicit feature expansion (subsequences of length $n$)
- Automatically defines the similarity measure (dot product)
- Need to devise a mechanism to calculate the kernel
What we’re essentially doing is...

- Defining our discrete object (String)
- Defining implicit feature expansion (subsequences of length $n$)
- Automatically defines the similarity measure (dot product)
- Need to devise a mechanism to calculate the kernel
What we’re essentially doing is...

- Defining our discrete object (string)
- Defining implicit feature expansion (subsequences of length $n$)
- Automatically defines the similarity measure (dot product)
- Need to devise a mechanism to calculate the kernel
What we’re essentially doing is...

- Defining our discrete object (string)
- Defining implicit feature expansion (subsequences of length $n$)
- Automatically defines the similarity measure (dot product)
- Need to devise a mechanism to calculate the kernel

Dynamic Programming
Applications

Relation Extraction

Semantic Role Labeling

Parsing and Tagging
Applications


Semantic Role Labeling

Parsing and Tagging
Applications

Relation Extraction


Semantic Role Labeling


Parsing and Tagging
### Applications

**Relation Extraction**

**Semantic Role Labeling**

**Parsing and Tagging**
Other Papers

- Theoretical foundations of Convolution Kernels
Other Papers

- Theoretical foundations of Convolution Kernels

- Tree Kernels for NLP

- People
  - Michael Collins – MIT
  - Alessandro Moschitti – University of Trento
  - Min Zhang, Goudong Zhou – Institute for Infocomm Research
About 7 meetings, 2 papers/meeting: 14 papers
About 7 meetings, 2 papers/meeting: 14 papers
Time: 12-1pm versus 1-2pm?
Issues/Questions

- About 7 meetings, 2 papers/meeting: 14 papers
- Time: 12-1pm versus 1-2pm?
- How do we volunteer (During meetings, email, wiki)?