PAC-Learning Unambiguous $k, l$-NTS $\leq$ Languages

Franco M. Luque  Gabriel Infante-Lopez

Grupo de Procesamiento de Lenguaje Natural
Universidad Nacional de Córdoba & CONICET
Córdoba, Argentina

ICGI 2010 – Valencia, Spain
September 16, 2010
Introduction

Our Goal

Find learnable classes of languages that include in some degree the natural languages.
Introduction

Our Goal

Find learnable classes of languages that include in some degree the natural languages.

Non Terminally Separated (NTS) languages (Clark, 2006)

- Unambiguous NTS (UNTS) languages: PAC-learnable.
- Natural language: not strictly NTS but close to it.
Introduction

Our Goal

Find learnable classes of languages that include in some degree the natural languages.

Non Terminally Separated (NTS) languages (Clark, 2006)

- Unambiguous NTS (UNTS) languages: PAC-learnable.
- Natural language: not strictly NTS but close to it.

$k, l$-NTS languages

- Adds a “fixed size context” notion to NTS, common in NL.
- A hierarchy of classes generalizing the NTS class.
- Analog to $k, l$-substitutable generalizing substitutable languages (Yoshinaka, 2008).
Introduction

$k, l$-NTS languages

- Fix an anomaly of $k, l$-NTS: Also consider the edges of sentences as valid contexts.
- More suitable for natural language modeling.
- A hierarchy of subclasses of the $k, l$-NTS classes.
**Introduction**

$k, l$-NTS$\leq$ languages

- Fix an anomaly of $k, l$-NTS: Also consider the edges of sentences as valid contexts.
- More suitable for natural language modeling.
- A hierarchy of subclasses of the $k, l$-NTS classes.

**PAC-Learnability Result**

- $k, l$-UNTS$\leq$ languages can be converted injectively to UNTS languages over a richer alphabet.
- So, they can be PAC-learned reusing Clark's learning algorithm for UNTS languages.
Outline of the Talk

1. Introduction
2. Definitions and Properties
3. Learning Algorithm for $k$, $l$-UNTS$\leq$ Languages
4. Proof of PAC-Learnability
5. Discussion
Non Terminally Separated (NTS) Languages

Informal (and Incomplete) Definition

- CFG where a string is always a *constituent* or it is always not.
- This is, there is a global set of constituents.
Informal (and Incomplete) Definition

- CFG where a string is always a *constituent* or it is always not.
- This is, there is a global set of constituents.

Example

The constituents are $C = \{ab, aa, aba, abb, baa\}$.

The distituents are $D = \{a, b, ba, bb\}$. 
Non Terminally Separated (NTS) Languages

**Informal (and Incomplete) Definition**

- CFG where a string is always a *constituent* or it is always not.
- This is, there is a global set of constituents.

**Example**

- The constituents are $C = \{ab, aa, aba, abb, baa\}$.
- The distituents are $D = \{a, b, ba, bb\}$.
Non Terminally Separated (NTS) Languages

Informal (and Incomplete) Definition

- CFG where a string is always a *constituent* or it is always not.
- This is, there is a global set of constituents.

Example

The constituents are $C = \{ab, aa, aba, abb, baa\}$.

The distituents are $D = \{a, b, ba, bb\}$.
Non Terminally Separated (NTS) Languages

Informal (and Incomplete) Definition

- CFG where a string is always a *constituent* or it is always not.
- This is, there is a global set of constituents.

Example

[Diagram of CFG with constituent and distituent sets]

- The constituents are \( C = \{ab, aa, aba, abb, baa\} \). (this is L)
- The distituents are \( D = \{a, b, ba, bb\} \).

Franco M. Luque
Non Terminally Separated (NTS) Languages

Informal (and Incomplete) Definition
- CFG where a string is always a constituent or it is always not.
- This is, there is a global set of constituents.

Example

```
  S
 / \  
X1  a  X2
  /\
 a  b  a
 / \  /  
 a  b  a

The constituents are $C = \{ab, aa, aba, abb, baa\}$. (find them!)

The distituents are $D = \{a, b, ba, bb\}$. (find them!)
```
$k, l$-Non Terminally Separated ($k, l$-NTS) Languages

Intuition

- Introduce the influence of fixed size contexts in the constituency decision.
- Different set of constituents for each possible context.
$k, l$-Non Terminally Separated ($k, l$-NTS) Languages

**Intuition**

- Introduce the influence of fixed size contexts in the constituency decision.
- Different set of constituents for each possible context.

**Example: A 0, 1-NTS grammar**

The constituents are $C_{\lambda, c} = \{ab, bd\}$, $C_{\lambda, d} = \emptyset$,  

The distituents are $D_{\lambda, c} = \{b, d\}$, $D_{\lambda, d} = \{ab, b\}$,  

Strings with smaller contexts are not affected.
$k$, $l$-Non Terminally Separated ($k$, $l$-NTS) Languages

**Intuition**

- Introduce the influence of fixed size contexts in the constituency decision.
- Different set of constituents for each possible context.

**Example: A 0, 1-NTS grammar**

- The constituents are $C_{\lambda,c} = \{ab, bd\}$, $C_{\lambda,d} = \emptyset$, . . .
- The distituents are $D_{\lambda,c} = \{b, d\}$, $D_{\lambda,d} = \{ab, b\}$, . . .
- Strings with smaller contexts are not affected.
Intuition

- Introduce the influence of fixed size contexts in the constituency decision.
- Different set of constituents for each possible context.

Example: A 0, 1-NTS grammar

- The constituents are $C_{\lambda, c} = \{ ab, bd \}$, $C_{\lambda, d} = \emptyset$, ....
- The distituents are $D_{\lambda, c} = \{ b, d \}$, $D_{\lambda, d} = \{ ab, b \}$, ....
- Strings with smaller contexts are not affected.
**Intuition**

- Introduce the influence of fixed size contexts in the constituency decision.
- Different set of constituents for each possible context.

**Example: A 0, 1-NTS grammar**

The constituents are $C_{\lambda,c} = \{ab, bd\}$, $C_{\lambda,d} = \emptyset$, $D_{\lambda,c} = \{b, d\}$, $D_{\lambda,d} = \{ab, b\}$, $\ldots$.

Strings with smaller contexts are not affected.
Intuition

- $k, l$-NTS $\leq$ grammars also involve strings occurrences that only have contexts smaller than $(k, l)$.
- Add prefix $\cdot^k$ and suffix $\cdot^l$ to every element of the language to have always contexts of size $(k, l)$ (define in terms of $k, l$-NTS).
$k$, $l$-Non Terminally Separated $\leq (k, l$-NTS$\leq$) Languages

Intuition

- $k$, $l$-NTS$\leq$ grammars also involve strings occurrences that only have contexts smaller than $(k, l)$.
- Add prefix $\bullet^k$ and suffix $\bullet^l$ to every element of the language to have always contexts of size $(k, l)$ (define in terms of $k$, $l$-NTS).

Example: A 0, 1-NTS$\leq$ grammar

- More sets of constituents: $C_{\lambda, \bullet} = \{bd, abc, bdc, abd\}$. 
Properties of $k, l$-NTS and $k, l$-NTS\leq$ Languages

**Properties**

- $0, 0$-NTS$\leq$ = $0, 0$-NTS = NTS.
- $k, l$-NTS conform a strict hierarchy:
  
  
  NTS
  →
  1, 0-NTS 0, 1-NTS
  →
  2, 0-NTS 1, 1-NTS 0, 2-NTS
  →
  ...
  ...
  ...
  
  Also does $k, l$-NTS$\leq$.
- $k, l$-NTS$\leq$ $\subseteq$ $k, l$-NTS, properly if $k + l > 0$. 
Learning Algorithm for $k, l$-UNTS≤ Languages

Intuition

- $k, l$-UNTS≤ languages can be injectively converted into UNTS languages.
- Just add the context information into the strings themselves.
- Example with $(k, l) = (1, 1)$: $abc$ is converted to $\bullet ab a bc b c \bullet$. 
Learning Algorithm for $k, l$-UNTS$\leq$ Languages

Intuition

- $k, l$-UNTS$\leq$ languages can be injectively converted into UNTS languages.
- Just add the context information into the strings themselves.
- Example with $(k, l) = (1, 1)$: $abc$ is converted to $\bullet ab a b_c b c \bullet$.
Learning Algorithm for \( k, l \)-UNTS\( \leq \) Languages

**Intuition**

- \( k, l \)-UNTS\( \leq \) languages can be injectively converted into UNTS languages.
- Just add the context information into the strings themselves.
- Example with \((k, l) = (1, 1)\): \(abc\) is converted to \(\bullet ab a b c b c \bullet\).

**k,l-PACCFG algorithm**

- **Input:** A sample \( S \). \( k, l \) and some other parameters.
- **Result:** A context-free grammar \( \hat{G} \).
- **Steps:**
  1. Convert \( S \) into a new sample \( S' \) by marking the contexts.
  2. Run PACCFG with \( S' \) and let \( \hat{G}' \) be the resulting grammar.
  3. Remove the marks in \( \hat{G}' \) and return the resulting grammar \( \hat{G} \).
Towards a Proof of PAC-Learnability

Intuition

- Marking contexts in a $k, l$-UNTS$\leq$ language gives a UNTS language.
- Given a $k, l$-UNTS$\leq$ grammar, modify its rules to build a UNTS grammar for the converted language.
  - Mark all the boundaries: bottom-up procedure.
  - Mark all the contexts: top-down procedure.

Example: 1, 1-NTS$\leq$

```
S
  |   |
X   c
  |   |
a   b
```
```
S
  |   |
X   c
  |   |
b   d
```
```
S
  |   |
a   Y
  |   |
b   d
```
Towards a Proof of PAC-Learnability

Intuition

- Marking contexts in a $k, l$-UNTS$\leq$ language gives a UNTS language.
- Given a $k, l$-UNTS$\leq$ grammar, modify its rules to build a UNTS grammar for the converted language.
  - Mark all the boundaries: bottom-up procedure.
  - Mark all the contexts: top-down procedure.

Example: 1, 1-NTS$\leq$

```
S
  X
   ? C
  a b
```
```
S
  X
   ? C
  b d
```
```
S
 a Y
 b d
```
Towards a Proof of PAC-Learnability

Intuition

- Marking contexts in a $k, l$-UNTS$\leq$ language gives a UNTS language.
- Given a $k, l$-UNTS$\leq$ grammar, modify its rules to build a UNTS grammar for the converted language.
  - Mark all the boundaries: bottom-up procedure.
  - Mark all the contexts: top-down procedure.

Example: $1, 1$-NTS$\leq$

```
S
 / \ 
X? ? c
/ \ 
| a | b |
```
```
S
 / \ 
X? ? c
/ \ 
| b | d |
```
```
S
 / 
| a | Y |
```
`
Towards a Proof of PAC-Learnability

Intuition

- Marking contexts in a $k, l$-UNTS$\leq$ language gives a UNTS language.
- Given a $k, l$-UNTS$\leq$ grammar, modify its rules to build a UNTS grammar for the converted language.
  - Mark all the boundaries: bottom-up procedure.
  - Mark all the contexts: top-down procedure.

Example: 1, 1-NTS$\leq$

\[
\begin{align*}
S & \quad S \\
\quad aXb & \quad bXd \\
\qquad a & \quad \quad b \\
\quad b & \quad d
\end{align*}
\]
Towards a Proof of PAC-Learnability

Intuition

- Marking contexts in a $k, l$-UNTS$\leq$ language gives a UNTS language.
- Given a $k, l$-UNTS$\leq$ grammar, modify its rules to build a UNTS grammar for the converted language.
  - Mark all the boundaries: bottom-up procedure.
  - Mark all the contexts: top-down procedure.

Example: $1, 1$-NTS$\leq$

```
  aS^c
   aX  b   c
  a   b      c

  bS^c
   bX  d   c
  b   d      c

  aS^d
   a   bY  d
  a   b       d
```
Towards a Proof of PAC-Learnability

Intuition

- Marking contexts in a $k, l$-$\text{UNTS} \leq$ language gives a UNTS language.
- Given a $k, l$-$\text{UNTS} \leq$ grammar, modify its rules to build a UNTS grammar for the converted language.
  - Mark all the boundaries: bottom-up procedure.
  - Mark all the contexts: top-down procedure.

Example: 1, 1-$\text{NTS} \leq$

\[ S \rightarrow aSc | aXb | a \]
\[ S \rightarrow bSc | bXd | b \]
\[ S \rightarrow aSd | a \]
\[ S \rightarrow bYd | b \]
Towards a Proof of PAC-Learnability

**Intuition**

- Marking contexts in a $k, l$-UNTS\(\leq\) language gives a UNTS language.
- Given a $k, l$-UNTS\(\leq\) grammar, modify its rules to build a UNTS grammar for the converted language.
  - Mark all the boundaries: bottom-up procedure.
  - Mark all the contexts: top-down procedure.

**Example: 1, 1-NTS\(\leq\)**

```
  S
 / \  /
S\ S
 /   /
aX b   bX d
|   /  |
X   Y
| a, b |
| b, d |
```
Towards a Proof of PAC-Learnability

Intuition

- Marking contexts in a $k, l$-UNTS$\leq$ language gives a UNTS language.
- Given a $k, l$-UNTS$\leq$ grammar, modify its rules to build a UNTS grammar for the converted language.
  - Mark all the boundaries: bottom-up procedure.
  - Mark all the contexts: top-down procedure.

Example: 1, 1-NTS$\leq$

\[
\begin{align*}
S & \rightarrow aSc \\
   & \quad \rightarrow aXb \quad \rightarrow a \quad \rightarrow a \ b \\
S & \rightarrow bSc \\
   & \quad \rightarrow bXd \quad \rightarrow b \quad \rightarrow b \ d \\
S & \rightarrow aSd \\
   & \quad \rightarrow aYd \quad \rightarrow b \quad \rightarrow b \ d
\end{align*}
\]
Towards a Proof of PAC-Learnability

Intuition

- Marking contexts in a $k, l$-UNTS$\leq$ language gives a UNTS language.
- Given a $k, l$-UNTS$\leq$ grammar, modify its rules to build a UNTS grammar for the converted language.
  - Mark all the boundaries: bottom-up procedure.
  - Mark all the contexts: top-down procedure.

Example: 1, 1-NTS$\leq$

```
S
  aS\ c
    aX\ b
      a\ b
      a\ b\ c
    b\ c

S
  bS\ c
    bX\ d
      b\ d
      b\ d\ c
    d\ c

S
  aSa
    a\ b
    a\ b\ d
    a\ b\ d\ c
  b\ Y\ d
    b\ a
    b\ d
    b\ d\ c
```
### Parameters and Bounds

#### Parameters

- Confidence $\delta$ and precision $\epsilon$.
- Alphabet $\Sigma$: converted to $\Sigma^k \times \Sigma \times \Sigma^l$, where $\Sigma_\bullet = \Sigma \cup \{\bullet\}$. 

---

Franco M. Luque
Parameters and Bounds

Parameters

- Confidence $\delta$ and precision $\epsilon$.
- Alphabet $\Sigma$: converted to $\Sigma^k \times \Sigma \times \Sigma^l$, where $\Sigma_\bullet = \Sigma \cup \{\bullet\}$.

Upper Bounds

- Number of non-terminals $n$: converted to $n|\Sigma|^{2(k+l)}$.
- Productions $p$: converted to $p|\Sigma|^{(k+l)(o+1)}$.
- Length of right sides $m$: doesn’t change.
- Expected number of substrings $L$: doesn’t change.
- Number of non-terminals in a right side $o$: new ($o \leq m$).
Parameters and Bounds

Parameters

- Confidence $\delta$ and precision $\epsilon$.
- Alphabet $\Sigma$: converted to $\Sigma^k \times \Sigma \times \Sigma^l$, where $\Sigma_\cdot = \Sigma \cup \{\cdot\}$.

Upper Bounds

- Number of non-terminals $n$: converted to $n|\Sigma|^{2(k+l)}$.
- Productions $p$: converted to $p|\Sigma|^{(k+l)(o+1)}$.
- Length of right sides $m$: doesn’t change.
- Expected number of substrings $L$: doesn’t change.
- Number of non-terminals in a right side $o$: new ($o \leq m$).

Assume that the underlying converted PCFG is $\mu_1$-distinguishable, $\mu_2$-reachable and $\nu$-separable with known values.
Proof of PAC-Learnability

Strategy

- Convert $k, l$-UNTS$\leq$ languages to $k, l - 1$-UNTS$\leq$:
  1. Left Marked Form: Formalization of the bottom-up procedure.
  2. Right Contextualized Grammar: Formalization of the top-down procedure.

- Extend this procedure to PCFGs preserving distributions.

- Apply induction and symmetry to convert $k, l$-UNTS$\leq$ languages to $0, 0$-UNTS$\leq$ =UNTS.

- Use this to prove that $k,l$-PACC[cell removed] is PAC over $k, l$-UNTS$\leq$ languages.
The Theorems

**Theorem: Conversion to UNTS PCFGs**

Let $G$ be a $k, l$-UNTS$\leq$ PCFG. Then, there is a UNTS PCFG $G'$ such that

1. $L(G') = \text{mark}_{k,l}(L(G))$, 
2. and for every $s \in L(G)$, $P_G(s) = P_{G'}(\text{mark}_{k,l}(s))$. 

Franco M. Luque
The Theorems

**Theorem: Conversion to UNTS PCFGs**

Let $G$ be a $k, l$-UNTS $\leq$ PCFG. Then, there is a UNTS PCFG $G'$ such that

1. $L(G') = \text{mark}_{k,l}(L(G))$,
2. and for every $s \in L(G)$, $P_G(s) = P_{G'}(\text{mark}_{k,l}(s))$.

**Theorem: PAC-learning**

Given $\delta$ and $\epsilon$, there is $N$ such that, if $S$ is a sample of a $k, l$-UNTS $\leq$ PCFG $G$ with $|S| > N$, then with probability greater than $1 - \delta$, $\hat{G} = k,l$-PACCFCG$(S)$ is such that

1. $L(\hat{G}) \subseteq L(G)$, and
2. $P_G(L(G) - L(\hat{G})) < \epsilon$. 

Franco M. Luque
Complexity

- Sample complexity:

\[
O \left( \frac{n' + p'}{\epsilon \mu_1 \mu_2 \nu^2} \right) = O \left( \frac{n|\Sigma|^{2(k+l)} + p|\Sigma|^{(k+l)(o+1)}}{\epsilon \mu_1 \mu_2 \nu^2} \right).
\]

- We directly assume known values of \( \mu_1, \nu \) and \( \mu_2 \) for the converted UNTS grammars.
### Complexity

- **Sample complexity:**

  \[
  O \left( \frac{n' + p'}{\epsilon \mu_1^m \mu_2^2 \nu^2} \right) = O \left( \frac{n|\Sigma|^{2(k+l)} + p|\Sigma|^{(k+l)(o+1)}}{\epsilon \mu_1^m \mu_2^2 \nu^2} \right).
  \]

- We directly assume known values of \( \mu_1, \nu \) and \( \mu_2 \) for the converted UNTS grammars.

- We could not use our “reduction” approach to prove PAC-learnability of all the \( k, l \)-UNTS languages.

- \( k, l \)-UNTS \( \leq \) are of more interest to us than \( k, l \)-UNTS.

- \( 1, 1 \)-UNTS \( \leq \) has high \( F_1 \) upper bound for WSJ10 over POS tags (96%) (Luque and Infante-Lopez, 2009 & 2010).