String Extension Learning

Jeffrey Heinz
heinz@udel.edu

University of Delaware

Association for Computational Linguistics
Uppsala, Sweden
July 13, 2010

How can something learn?

1. How do people generalize beyond their experience?
2. How can any*thing* that computes generalize beyond its experience?

- Linguistics
- Computer Science
- Artificial Intelligence
- Natural Language Processing
- Psychology
- Language Acquisition
- Philosophy
- ...

This talk

1. Provide a recipe for constructing classes of languages (families of possible generalizations)

$$f \to \mathcal{L}_f$$

2. Show each these classes are identifiable in the limit from positive data (Gold 1967) with learners that are *incremental, globally consistent, locally conservative*, and *set-driven.*
3. Reveal the lattice structure underlying this class of languages and how the learner "climbs the lattice" (Kasprzik and Koetzing 2010)

This talk

1. Provide a recipe for constructing classes of languages (families of possible generalizations)

$$f \to \mathcal{L}_f$$

2. Show each these classes are identifiable in the limit from positive data (Gold 1967) with learners that are *incremental*, *globally consistent*, *locally conservative*, and *set-driven*.
3. Reveal the lattice structure underlying this class of languages and how the learner "climbs the lattice" (Kasprzik and Koetzing 2010)

Each string in the language maps to part of the grammar which generates a set of strings in the language

String Extension Learners: Advantages

1. Distribution-free learning is a more difficult learning criteria than non-distribution-free learning criteria (Gold 1967, Horning 1969, Valiant 1984, Angluin 1988, Blumer et al. 1989)
2. Learners are very simple (useful pedagogically).
3. Learners are efficient (in size of learning sample) if f is efficient in the size of the word.
4. String extension learnable classes include ones of infinite size and ones which contain context-sensitive languages.

String Extension Learners: Advantages

5. Provides a unified learning-theoretic analysis of many previously discussed learnable classes.

Locally k-Testable	Piecewise k-Testable
Strictly k-Local	Strictly k-Piecewise
Strongly Testable	Definite

   (McNaughton and Papert 1971, Rogers and Pullum, to appear, Simon 1975, Rogers et al., 2009, Beauquier and Pin 1991, Brzozowski 1962)

6. Probabilistic versions of many of these classes exist (e.g. previous talk!)
7. Different perspective: modular learning

Identification in the Limit from Positive Data

- A *text* t is an infinite sequence: $t(0), t(1), \ldots$ with each $t(i) \in \Sigma^* \cup \{\#\}$. The $\#$ is a *pause*.
- $t[i]$ denotes the finite sequence $t(0), t(1), \ldots t(i)$.
- The *content* of text t is the set $\{t(i) : i \in \mathbb{N}\}$.
- t is a *positive text for a language* L iff $content(t) = L$.
- Following Jain et al. 1999, let SEQ be the set of all possible $t[i]$.
- A *learner* is a function $\phi : SEQ \rightarrow \mathcal{G}$. The elements of $\mathcal{G}$ generate languages in some well-defined way.

Identification in the Limit from Positive Data

- A learner *converges on a text* t iff there exists $i \in \mathbb{N}$ and a grammar G such that for all $j > i$, $\phi(t[j]) = G$.
- A learner ϕ *identifies a language* L *in the limit* iff for any positive text t for L, ϕ converges on t to grammar G and $L(G) = L$.
- A learner ϕ *identifies a class of languages* $\mathcal{L}$ *in the limit* iff for any $L \in \mathcal{L}$, ϕ identifies L in the limit.

An Example of String Extension Learning

k-factor languages:
A word is well-formed iff the contiguous subsequences in the word are well-formed

Functions

$$fac_k(w) = \{x \in \Sigma^k : \exists u, v \in \Sigma^* \text{ such that } w = uxv\} \tag{1}$$

Grammars

$$G \in \mathcal{P}(\Sigma^k) \tag{2}$$

Languages

$$L(G) = \{w : fac_k(w) \subseteq G\} \tag{3}$$

Example: Illustration of k-factor Learning

Let $L = \Sigma^* \backslash \Sigma^* ba \Sigma^*$

time	Word w	$fac_2(w)$	Grammar G	Language of G
-1			$\emptyset$	$\emptyset$
0	$aaaa$	$\{aa\}$	$\{\mathbf{aa}\}$	a^*
1	aab	$\{aa,\ ab\}$	$\{aa,\ \mathbf{ab}\}$	$a^* \cup a^*b$
2	a	$\emptyset$	$\{aa,\ ab\}$	$a^* \cup a^*b$
3	bbb	$\{bb\}$	$\{aa,\ ab,\ \mathbf{bb}\}$	$\Sigma^* \backslash \Sigma^* ba \Sigma^*$
4	$abbb$	$\{ab,\ bb\}$	$\{aa,\ ab,\ bb\}$	$\Sigma^* \backslash \Sigma^* ba \Sigma^*$
...				

Example: Illustration of k-factor Learning

Let $L = \Sigma^* \backslash \Sigma^* ba \Sigma^*$

time	Word w	$fac_2(w)$	Grammar G	Language of G
-1			$\emptyset$	$\emptyset$
0	$aaaa$	$\{aa\}$	$\{\mathbf{aa}\}$	a^*
1	aab	$\{aa,\ ab\}$	$\{aa,\ \mathbf{ab}\}$	$a^* \cup a^*b$
2	a	$\emptyset$	$\{aa,\ ab\}$	$a^* \cup a^*b$
3	bbb	$\{bb\}$	$\{aa,\ ab,\ \mathbf{bb}\}$	$\Sigma^* \backslash \Sigma^* ba \Sigma^*$
4	$abbb$	$\{ab,\ bb\}$	$\{aa,\ ab,\ bb\}$	$\Sigma^* \backslash \Sigma^* ba \Sigma^*$
...				

Example: Illustration of k-factor Learning

Let $L = \Sigma^* \backslash \Sigma^* ba \Sigma^*$

time	Word w	$fac_2(w)$	Grammar G	Language of G
-1			$\emptyset$	$\emptyset$
0	$aaaa$	$\{aa\}$	$\{\mathbf{aa}\}$	a^*
1	aab	$\{aa,\ ab\}$	$\{aa,\ \mathbf{ab}\}$	$a^* \cup a^*b$
2	a	$\emptyset$	$\{aa,\ ab\}$	$a^* \cup a^*b$
3	bbb	$\{bb\}$	$\{aa,\ ab,\ \mathbf{bb}\}$	$\Sigma^* \backslash \Sigma^* ba \Sigma^*$
4	$abbb$	$\{ab,\ bb\}$	$\{aa,\ ab,\ bb\}$	$\Sigma^* \backslash \Sigma^* ba \Sigma^*$
...				

Example: Illustration of k-factor Learning

Let $L = \Sigma^* \backslash \Sigma^* ba \Sigma^*$

time	Word w	$fac_2(w)$	Grammar G	Language of G
-1			$\emptyset$	$\emptyset$
0	$aaaa$	$\{aa\}$	$\{\mathbf{aa}\}$	a^*
1	aab	$\{aa,\ ab\}$	$\{aa,\ \mathbf{ab}\}$	$a^* \cup a^*b$
2	a	$\emptyset$	$\{aa,\ ab\}$	$a^* \cup a^*b$
3	bbb	$\{bb\}$	$\{aa,\ ab,\ \mathbf{bb}\}$	$\Sigma^* \backslash \Sigma^* ba \Sigma^*$
4	$abbb$	$\{ab,\ bb\}$	$\{aa,\ ab,\ bb\}$	$\Sigma^* \backslash \Sigma^* ba \Sigma^*$
...				

Example: Illustration of k-factor Learning

Let $L = \Sigma^* \backslash \Sigma^* ba \Sigma^*$

time	Word w	$fac_2(w)$	Grammar G	Language of G
-1			$\emptyset$	$\emptyset$
0	$aaaa$	$\{aa\}$	$\{\mathbf{aa}\}$	a^*
1	aab	$\{aa,\ ab\}$	$\{aa,\ \mathbf{ab}\}$	$a^* \cup a^*b$
2	a	$\emptyset$	$\{aa,\ ab\}$	$a^* \cup a^*b$
3	bbb	$\{bb\}$	$\{aa,\ ab,\ \mathbf{bb}\}$	$\Sigma^* \backslash \Sigma^* ba \Sigma^*$
4	$abbb$	$\{ab,\ bb\}$	$\{aa,\ ab,\ bb\}$	$\Sigma^* \backslash \Sigma^* ba \Sigma^*$
...				

Example: Illustration of k-factor Learning

Let $L = \Sigma^* \backslash \Sigma^* ba \Sigma^*$

time	Word w	$fac_2(w)$	Grammar G	Language of G
-1			$\emptyset$	$\emptyset$
0	$aaaa$	$\{aa\}$	$\{\mathbf{aa}\}$	a^*
1	aab	$\{aa,\ ab\}$	$\{aa,\ \mathbf{ab}\}$	$a^* \cup a^*b$
2	a	$\emptyset$	$\{aa,\ ab\}$	$a^* \cup a^*b$
3	bbb	$\{bb\}$	$\{aa,\ ab,\ \mathbf{bb}\}$	$\Sigma^* \backslash \Sigma^* ba \Sigma^*$
4	$abbb$	$\{ab,\ bb\}$	$\{aa,\ ab,\ bb\}$	$\Sigma^* \backslash \Sigma^* ba \Sigma^*$
...				

What matters?

k-factor languages:
A word is well-formed iff the contiguous subsequences in the word are well-formed

Functions

$$fac_k(w) = \{x \in \Sigma^k : \exists u, v \in \Sigma^* \text{ such that } w = uxv\} \tag{1}$$

Grammars

$$G \in \mathcal{P}(\Sigma^k) \tag{2}$$

Languages

$$L(G) = \{w : fac_k(w) \subseteq G\} \tag{3}$$

Definitions of String Extentsion Functions, Grammars, and Languages

- Let f be a total function with domain Σ^* and codomain the finite powerset of A, written $\mathcal{P}_{fin}(A)$.

Functions

$$f : \Sigma^* \rightarrow \mathcal{P}_{fin}(A) \tag{4}$$

Grammars

$$G \in \mathcal{P}_{fin}(A) \tag{5}$$

Languages

$$L_f(G) = \{w \in \Sigma^* : f(w) \subseteq G\} \tag{6}$$

Classes

$$\mathcal{L}_f = \{L(G) : G \in \mathcal{P}_{fin}(A)\} \tag{7}$$

Structure in $\mathcal{L}_f$

Theorem (Closure under intersection)

For any $f \in SEF$, $\mathcal{L}_f$ is closed under intersection.

- String extension language classes are not in general closed under union, complement or reversal (counterexamples are given later as examples.)

Extending domain of f

$$f(L) = \bigcup_{w \in L} f(w) \tag{8}$$

Lemma (Monotonicity)

Let $L, L' \in \mathcal{L}_f$. $L \subseteq L'$ *iff* $f(L) \subseteq f(L')$

Characteristic Sample for each $L \in \mathcal{L}_f$

Lemma (Smallest L in $\mathcal{L}_f$)

For any finite $L_0 \subseteq \Sigma^*$, $L = L(f(L_0))$ is the smallest language in $\mathcal{L}_f$ containing L_0.

Theorem (Characteristic Sample)

For all $L \in \mathcal{L}_f$, there is a finite sample S such that L is the smallest language in $\mathcal{L}_f$ containing S. S is called a characteristic sample of L in $\mathcal{L}_f$.

Corollary

For all $L \in \mathcal{L}_f$, a characteristic sample is $f(L)$.

The String Extension Learner

Definition

For all $f \in \mathcal{SEF}$, define ϕ_f as follows:

$$\phi_f(t[i]) = \begin{cases} \emptyset & \text{if } i = -1 \\ \phi_f(t[i-1]) & \text{if } t(i) = \# \\ \phi_f(t[i-1]) \cup f(t(i)) & \text{otherwise} \end{cases}$$

Results 1

- A learner ϕ is *maximally consistent* iff for each i, $content(t[i]) \subseteq L(\phi(t[i]))$.
- A learner ϕ is *locally conservative* iff whenever $\phi(t[i]) \neq \phi(t[i-1])$, it is the case that $t(i) \notin L(\phi([i-1]))$.

Lemma

ϕ_f is maximally consistent and locally conservative.

Lemma (set-driven)

For any text t and any $i \in \mathbb{N}$, $\phi(t[i]) = f(content(t[i]))$.

Results 2

Theorem (Identifiability)

If f is a string extension function, then ϕ_f identifies $\mathcal{L}_f$ in the limit.

Piecewise Testable in the Strict Sense
Rogers et al. 2009

A word is well-formed iff the subsequences
(not necessarily contiguous, and up to some length k)
in the word are well-formed

Definition

$u = a_1 \dots a_n$ ($a_i \in \Sigma$) is a subsequence of w iff there exists $v_0 \dots v_n \in \Sigma^*$ such that $w = v_0 a_1 \dots a_n v_n$. We write $u \sqsubseteq w$.

- The "Piecewise" function picks out the (not necessarily contiguous) subsequences up to length k

$$P_k(w) = \{u \in \Sigma^{\leq k} : u \sqsubseteq w\} \tag{9}$$

Piecewise Testable in the Strict Sense
Learning Illustration

Let $L = \Sigma^* \backslash \Sigma^* b \Sigma^* b \Sigma^*$

time	Word w	$P_2(w)$	Grammar G	Language of G
0	$aaaa$	$\{\epsilon, a, aa\}$	{ϵ, **a, aa**}	a^*
1	aab	$\{\epsilon, a, b, aa, ab\}$	{ϵ, a, aa, **b, ab**}	$a^* \cup a^*b$
2	baa	$\{\epsilon, a, b, aa, ba\}$	{ϵ, a, b, aa, ab **ba**}	$\Sigma^* \backslash (\Sigma^* b \Sigma^* b \Sigma^*)$
3	aba	$\{\epsilon, a, b, ab, ba\}$	{ϵ, a, b, aa, ab, ba}	$\Sigma^* \backslash (\Sigma^* b \Sigma^* b \Sigma^*)$
...				

Piecewise Testable in the Strict Sense
Learning Illustration

Let $L = \Sigma^* \backslash \Sigma^* b \Sigma^* b \Sigma^*$

time	Word w	$P_2(w)$	Grammar G	Language of G
0	$aaaa$	$\{\epsilon, a, aa\}$	{ϵ, **a, aa**}	a^*
1	aab	$\{\epsilon, a, b, aa, ab\}$	{ϵ, a, aa, **b, ab**}	$a^* \cup a^*b$
2	baa	$\{\epsilon, a, b, aa, ba\}$	{ϵ, a, b, aa, ab **ba**}	$\Sigma^* \backslash (\Sigma^* b \Sigma^* b \Sigma^*)$
3	aba	$\{\epsilon, a, b, ab, ba\}$	{ϵ, a, b, aa, ab, ba}	$\Sigma^* \backslash (\Sigma^* b \Sigma^* b \Sigma^*)$
...				

Piecewise Testable in the Strict Sense
Learning Illustration

Let $L = \Sigma^* \backslash \Sigma^* b \Sigma^* b \Sigma^*$

time	Word w	$P_2(w)$	Grammar G	Language of G
0	$aaaa$	$\{\epsilon, a, aa\}$	$\{\epsilon$**, a, aa**$\}$	a^*
1	aab	$\{\epsilon, a, b, aa, ab\}$	$\{\epsilon$, a, aa, **b, ab**$\}$	$a^* \cup a^* b$
2	baa	$\{\epsilon, a, b, aa, ba\}$	$\{\epsilon$, a, b, aa, ab **ba**$\}$	$\Sigma^* \backslash (\Sigma^* b \Sigma^* b \Sigma^*)$
3	aba	$\{\epsilon, a, b, ab, ba\}$	$\{\epsilon$, a, b, aa, ab, ba$\}$	$\Sigma^* \backslash (\Sigma^* b \Sigma^* b \Sigma^*)$
...				

Piecewise Testable in the Strict Sense
Learning Illustration

Let $L = \Sigma^* \backslash \Sigma^* b \Sigma^* b \Sigma^*$

time	Word w	$P_2(w)$	Grammar G	Language of G
0	$aaaa$	$\{\epsilon, a, aa\}$	{ϵ, **a, aa**}	a^*
1	aab	$\{\epsilon, a, b, aa, ab\}$	{ϵ, a, aa, **b, ab**}	$a^* \cup a^*b$
2	baa	$\{\epsilon, a, b, aa, ba\}$	{ϵ, a, b, aa, ab **ba**}	$\Sigma^* \backslash (\Sigma^* b \Sigma^* b \Sigma^*)$
3	aba	$\{\epsilon, a, b, ab, ba\}$	{ϵ, a, b, aa, ab, ba}	$\Sigma^* \backslash (\Sigma^* b \Sigma^* b \Sigma^*)$
...				

Piecewise Testable in the Strict Sense
Learning Illustration

Let $L = \Sigma^* \backslash \Sigma^* b \Sigma^* b \Sigma^*$

time	Word w	$P_2(w)$	Grammar G	Language of G
0	$aaaa$	$\{\epsilon, a, aa\}$	{ϵ, **a, aa**}	a^*
1	aab	$\{\epsilon, a, b, aa, ab\}$	{ϵ, a, aa, **b, ab**}	$a^* \cup a^* b$
2	baa	$\{\epsilon, a, b, aa, ba\}$	{ϵ, a, b, aa, ab **ba**}	$\Sigma^* \backslash (\Sigma^* b \Sigma^* b \Sigma^*)$
3	aba	$\{\epsilon, a, b, ab, ba\}$	{ϵ, a, b, aa, ab, ba}	$\Sigma^* \backslash (\Sigma^* b \Sigma^* b \Sigma^*)$
...				

Piecewise Testable in the Strict Sense
Learning Illustration

Let $L = \Sigma^* \backslash \Sigma^* b \Sigma^* b \Sigma^*$

time	Word w	$P_2(w)$	Grammar G	Language of G
0	$aaaa$	$\{\epsilon, a, aa\}$	{ϵ, **a, aa**}	a^*
1	aab	$\{\epsilon, a, b, aa, ab\}$	{ϵ, a, aa, **b, ab**}	$a^* \cup a^* b$
2	baa	$\{\epsilon, a, b, aa, ba\}$	{ϵ, a, b, aa, ab **ba**}	$\Sigma^* \backslash (\Sigma^* b \Sigma^* b \Sigma^*)$
3	aba	$\{\epsilon, a, b, ab, ba\}$	{ϵ, a, b, aa, ab, ba}	$\Sigma^* \backslash (\Sigma^* b \Sigma^* b \Sigma^*)$
...				

Applications of Piecewise Learning

Models of

- Phonotactic Learning (Heinz 2007, to appear)
- Reading Comprehension (Whitney 2001, Grainger and Whitney2004, Whitney and Cornelissen 2008)
- Text classification (Lodhi et al. 2001, Cancedda et al. 2003)
- ...

An infinite-sized class relating to the Parikh map (1966)

A word is well-formed iff the number of as
in the word belongs to some finite set of numbers

i	$t(i)$	$f_a(t(i))$	Grammar G	$L(G)$
0	$aaaa$	$\{4\}$	$\{\mathbf{4}\}$	B_4
1	$bbabbba$	$\{2\}$	$\{4,\ \mathbf{2}\}$	$B_4 \cup B_2$
2	$bbbaa$	$\{2\}$	$\{4,\ 2\}$	$B_4 \cup B_2$
3	$aaab^{100}$	$\{3\}$	$\{4,\ 2,\ \mathbf{3}\}$	$B_4 \cup B_2 \cup B_3$
...				

Table: Illustration of ϕ_{f_a} on a positive text of L({2, 3, 4}). B_i is the block of all and only those words containing exactly i as.

Build new string extension classes from others

Examples

1. Words are well-formed iff the contiguous subsequences of length 3 are well-formed and the (potentially discontiguous) subsequences up to length 2 are well formed.

$$f(w) = \langle fac_3(w), P_2(w) \rangle$$

2. Words are well-formed iff the contiguous subsequences of length k are well-formed and the number of as mod n is 0.

$$mod_n(w) = 1 \text{ iff } |w|_a/n = 0$$

$$f(w) = \langle fac_k(w), mod_n(w) \rangle$$

3. ...

Classes with context-free languages

$f(w) = 0$ iff $w \in a^n b^n$ and 1 otherwise

grammar G	Language of G
$\emptyset$	$\emptyset$
$\{0\}$	$a^n b^n$
$\{1\}$	$\Sigma^* \backslash a^n b^n$
$\{0,\ 1\}$	Σ^*

Table: The language class $\mathcal{L}_f$.

Classes with context-free languages

$$f(w) = 0 \text{ iff } w \in a^n b^n \text{ and } 1 \text{ otherwise}$$

grammar G	Language of G
$\emptyset$	$\emptyset$
$\{0\}$	$a^n b^n$
$\{1\}$	$\Sigma^* \backslash a^n b^n$
$\{0,\ 1\}$	Σ^*

Table: The language class $\mathcal{L}_f$.

The pattern languages (Angluin 1980), which include some non-context-free languages and which are infinite in size, are identifiable in the limit in this manner (Kasprzik and Koetzing 2010).

The function f partitions Σ^*

$$w_1 \sim w_2 \text{ iff } f(w_1) = f(w_2)$$

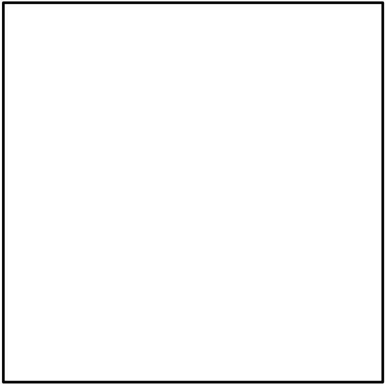


Figure: Σ^*

The function f partitions Σ^*

$$w_1 \sim w_2 \text{ iff } f(w_1) = f(w_2)$$

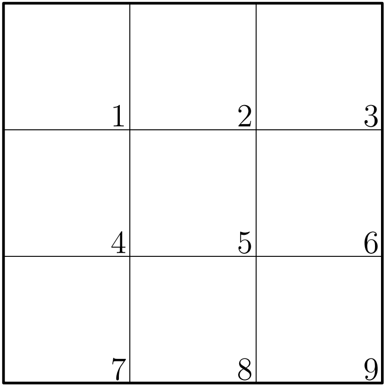


Figure: Example: a finite partition of Σ^* given by f.

Theorem

Every language $L \in \mathcal{L}_f$ *is a finite union of blocks of* π_f*.*

Corollary

$(\mathcal{L}_f, \subseteq)$ *is a lattice.*

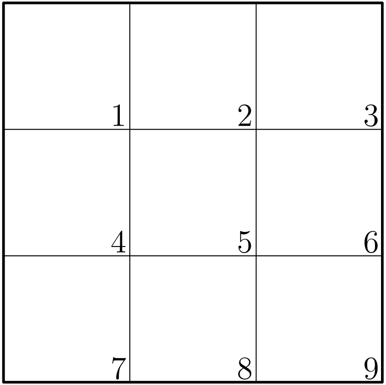


Figure: Examples of languages in $\mathcal{L}_f$.

Theorem

Every language $L \in \mathcal{L}_f$ is a finite union of blocks of π_f.

Corollary

$(\mathcal{L}_f, \subseteq)$ *is a lattice.*

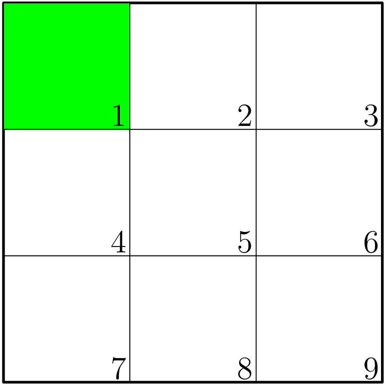


Figure: Examples of languages in $\mathcal{L}_f$.

Theorem

Every language $L \in \mathcal{L}_f$ is a finite union of blocks of π_f.

Corollary

$(\mathcal{L}_f, \subseteq)$ *is a lattice.*

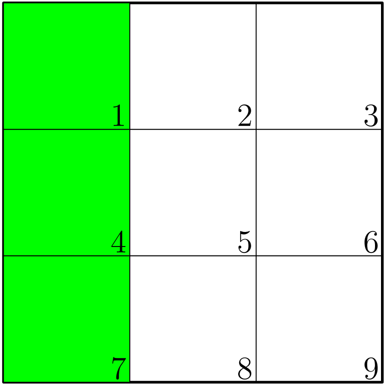


Figure: Examples of languages in $\mathcal{L}_f$.

Lattice structure of the hypothesis space (Kasprzik and Koetzing 2010)

- Let $(\mathcal{G}, \leq)$ be a lattice.
- Let f be a total function with domain Σ^* and codomain $\mathcal{G}$.

Functions

$$f : \Sigma^* \to \mathcal{G} \tag{10}$$

Grammars

$$G \in \mathcal{G} \tag{11}$$

Languages

$$L_f(G) = \{w \in \Sigma^* : f(w) \leq G\} \tag{12}$$

Classes

$$\mathcal{L}_f = \{L(G) : G \in \mathcal{G}\} \tag{13}$$

Learners are the same except replace $\cup$ with *least upper bound*.

Theorem (Isomorphism)

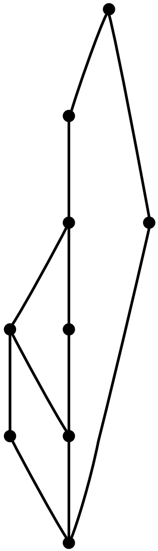
The lattice $(\mathcal{L}_f, \subseteq)$ is isomorphic to $(\mathcal{G}, \leq)$.

Example: Lattice of grammars

Each node represents a block in the partition of Σ^* given by f.

Each node N also represents a language. The language is all words in all blocks of all nodes dominated by N.

Each node also represents a grammar - a finite description of this potentially infinitely-sized language.

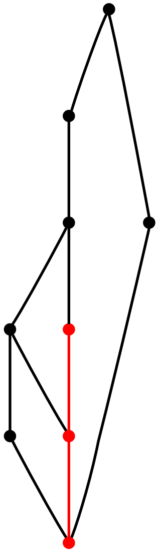


Example: Lattice of grammars

Each node represents a block in the partition of Σ^* given by f.

Each node N also represents a language. The language is all words in all blocks of all nodes dominated by N.

Each node also represents a grammar - a finite description of this potentially infinitely-sized language.

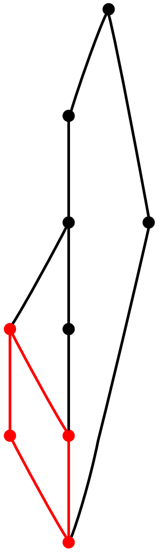


Example: Lattice of grammars

Each node represents a block in the partition of Σ^* given by f.

Each node N also represents a language. The language is all words in all blocks of all nodes dominated by N.

Each node also represents a grammar - a finite description of this potentially infinitely-sized language.

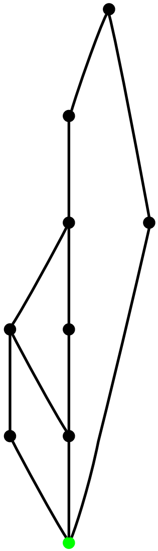


Example: Lattice of grammars

Learners can make inferences in two ways:

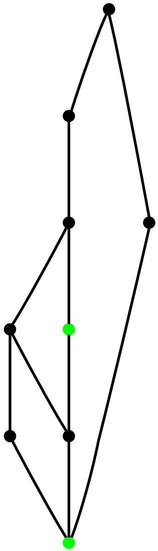
1. If a node is part of the language, everything below it is too.
2. If two nodes are part of the language, the least upper bound is too.

Assume the starting point is the least element in the example.



Example: Lattice of grammars

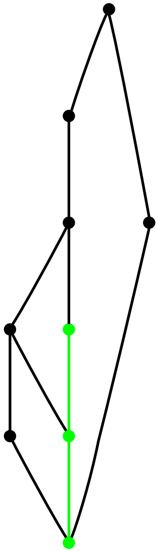
Suppose the learner observes w_1 and $f(w_1)$ maps to the node shown.



Example: Lattice of grammars

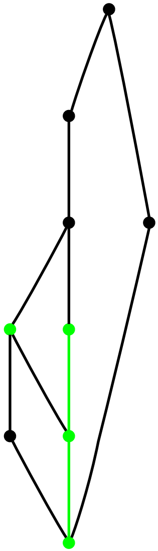
Suppose the learner observes w_1 and $f(w_1)$ maps to the node shown.

Then the learner can infer everything below that node is also in the language.



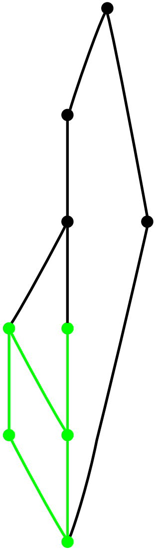
Example: Lattice of grammars

Suppose the learner then observes w_2 and $f(w_2)$ maps to this other node.



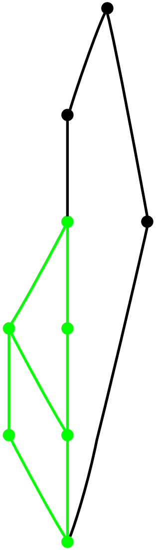
Example: Lattice of grammars

Then the learner can infer all words in blocks below that node are also in the language.



Example: Lattice of grammars

And the learner can infer words in the least upper bound are also in the language.



Summary

1. Different kinds of patterns can by learned by employing different kinds of learners.
2. String extension learners are simple and provably correct.
3. The structure present in such classes underlie some common models in NLP like n-gram models
4. However, the structure is independent of the dependence on conditioning well-formedness or likelihood on *contiguous subsequences*.
5. As long as grammars $\mathcal{G}$ are defined as finite subsets of some set A (or more generally as lattices $(\mathcal{G}, \leq)$), then a function $f : \Sigma^* \to \mathcal{G}$ defines a class of languages which identifies $\mathcal{L}_f$ in the limit from positive data.
6. We have a recipe for new learnable classes, for combining them, and a unified learning-theoretic analysis of many existing classes.

Future Work

1. *Hyperplane learning* (Clark 2006, Clark et al. 2006) and *function-distinguishable* learning (Fernau 2003) also associate language classes with a single function and show that they are learnable. It would be interesting to compare the learnable classes of languages and the learning strategies.
2. What kinds of of lattice-structured hypothesis spaces have finite VC dimension (i.e. are PAC-learnable)?
3. Learning classes of stochastic languages by "climbing the lattice"...

Future Work

1. *Hyperplane learning* (Clark 2006, Clark et al. 2006) and *function-distinguishable* learning (Fernau 2003) also associate language classes with a single function and show that they are learnable. It would be interesting to compare the learnable classes of languages and the learning strategies.
2. What kinds of of lattice-structured hypothesis spaces have finite VC dimension (i.e. are PAC-learnable)?
3. Learning classes of stochastic languages by "climbing the lattice"…

Thank You.