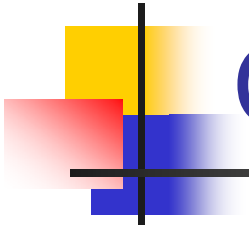
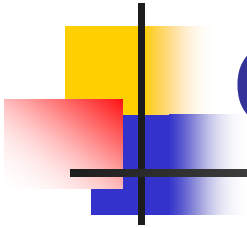
Computational Learning Theory
Learning and Mathematical Algorithms

Akihiro Yamamoto 山本 章博
http://www.iip.ist.i.kyoto-u.ac.jp/member/akihiro/
akihiro@i.kyoto-u.ac.jp



Contents

- GCD and the Euclidian Algorithm
- Math of Rings and Polynomials
- Hilbert's basis theorem
- Relation to Machine Learning



GCD and the Euclidian Algorithm

- In mathematics, the greatest common divisor (GCD) of two or more integers, which are not all zero, is the largest positive integer that divides each of the integers. For example, the gcd of 8 and 12 is 4. [Wikipedia]
- By the Euclidian algorithm, gcd(m, n) can be computed *efficiently*.

$$\gcd(m, 0) = m$$

$$\gcd(m, n) = \gcd(n, m \bmod n)$$

Example

gcd(34, 21)=gcd(21, 13)=gcd(13, 8)=gcd(8, 5)=gcd(5, 3)
=gcd(3, 2)=gcd(3, 2)=gcd(2, 1)=gcd(1, 1)=gcd(1, 0)=0

GCD and Learning

A class of languages in $\mathbf{N}$:

$\mathsf{C} = \{L(m) \mid m \in \mathbf{N}\}$

$L(m) = \{01\underbrace{1\ldots1}_{n}0 \mid n \bmod m = 0\}$

$L(m) = \{n \in \mathbf{N} \mid n \bmod m = 0\}$

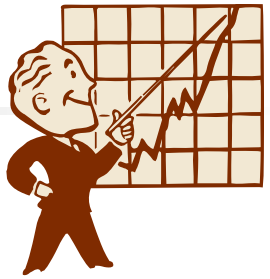
A class of languages in $\mathbf{Z}$:

$\mathsf{C} = \{L(m) \mid m \in \mathbf{N}\}$

$L(m) = \{\underbrace{1\ldots1}_{n} \mid n \bmod m = 0\} \cup \{01\underbrace{1\ldots1}_{n} \mid n \bmod m = 0\}$

$L(m) = \{n \in \mathbf{Z} \mid \mid n \mid \bmod m = 0\}$

GCD and Learning



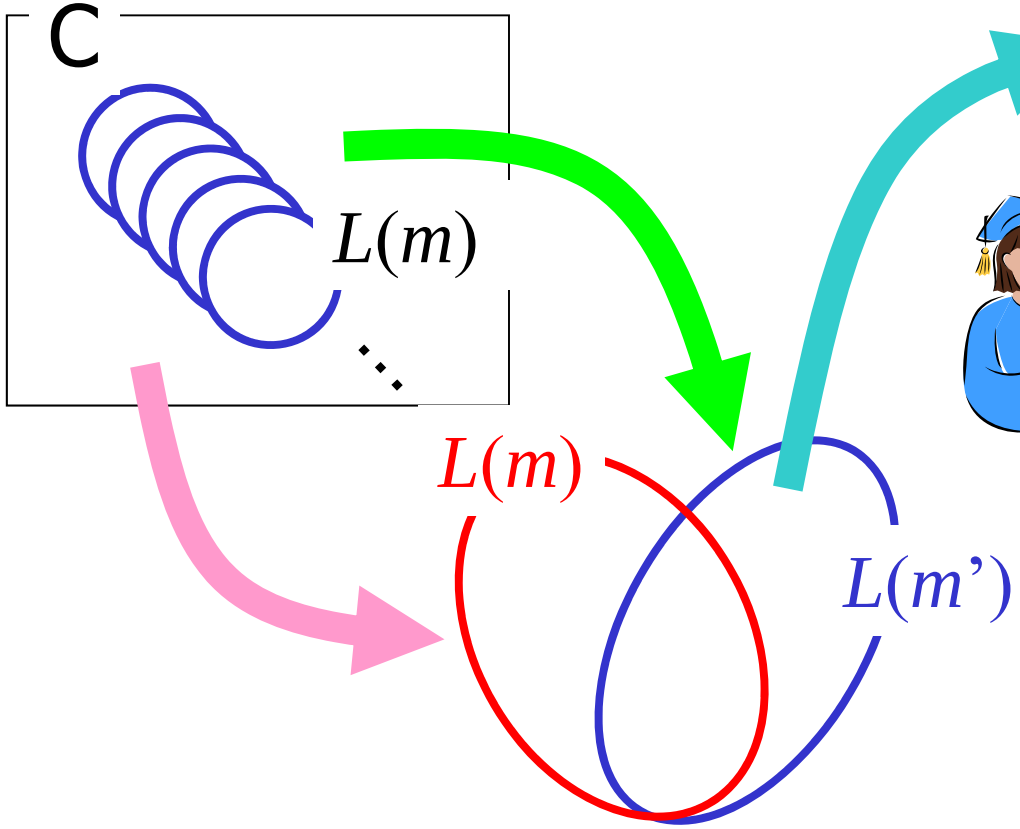
- A class of languages :

  $C = \{L(m) \mid m \in \mathbf{N}\}$

  $L(m) = \{01\ldots10 \mid n \bmod m = 0\}$

Positive presentation

72, 48, 60, …,12,…



Compute the GCD of $e_1, e_2, \ldots, e_k$ with the Euclidean Algorithm

Conjecture

72, 24, 12,…,12,…

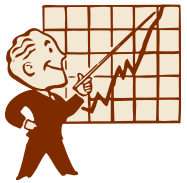
C4: Finite thickness

- A class C of languages has the finite thickness property if for all $w \in \Sigma^*$ only a finite number of languages contains w,

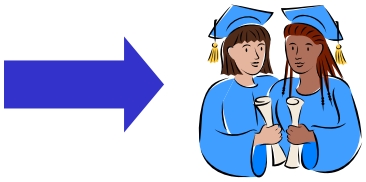
Theorem [Angluin] If a class C of languages has the finite thickness, C is identifiable in the limit from positive data.

Note : Even if C has the finite thickness property, $\cup^N U$ might not have the same property.

Identification in the limit [Gold]



$e_1, e_2, e_3, \ldots$



$g_1, g_2, g_3, \ldots$

- A learning algorithm A EX-identifies $L(g)$ in the limit from positive presentations if
  for any positive presentation $\sigma = e_1, e_2, e_3, \ldots$ of $L(g)$ and the output sequence $g_1, g_2, g_3, \ldots$ of A, there exists N such that for all $n > N$ $g_n = g'$ and $L(g') = L(g)$
- A learning algorithm A EX-identifies a class C of languages in the limit from positive presentations if
  A EX-identifies every language in C in the limit from positive presentations.

Proving that C is identifiable

- From the finite thickness condition:

  The class $\mathbf{C} = \{L(m) \mid m \in \mathbf{N}\}$ has the finite thickness property.

- From a property of natural numbers and the property

  $$\mathrm{GCD}(e_1, e_2, \ldots, e_k) \geq \mathrm{GCD}(e_1, e_2, \ldots, e_k, e_{k+1})$$

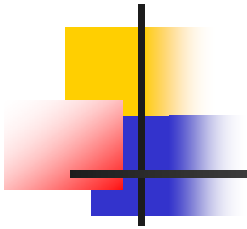
  The property is :

  Let $a_1, a_2, \ldots, a_n, \ldots$ be a infinite sequence of natural numbers satisfying that

  $$a_n \geq a_{n+1} \quad \text{for all } n \geq 1.$$

  Then there is $N \geq 1$ such that $a_n = a_{n+1}$ for all $n \geq N$.

Generalizing the Setting



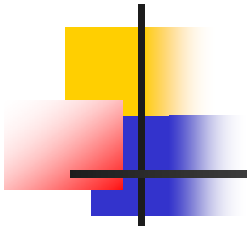
- The class $\mathsf{C} = \{L(m) \mid m \in \mathbf{N}\}$ is defined with multiplication
- The Euclidean Algorithm as the learning machine is based on

computing remainders of two integers,

which can be constructed of division and subtraction, and division of two integers can be implemented with multiplication and subtraction. So the Euclidean Algorithm can be implemented with

multiplication, subtraction (the inverse of addition).

Generalizing the Setting

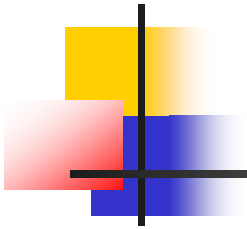


- The class $\mathsf{C} = \{L(m) \mid m \in \mathbf{N}\}$ is defined with multiplication
- The Euclidean Algorithm as the learning machine is based on

computing remainders of two integers,

which can be constructed of division and subtraction, and division of two integers can be implemented with multiplication and subtraction. So the Euclidean Algorithm can be implemented with

multiplication, subtraction (the inverse of addition).



Monomimial Case

- Consider the case $\mathsf{U} = \{x^m y^n \mid m, n \in \mathbf{N}\}$

  $L(x^k y^l) = \{ x^{k+a} y^{l+b} \mid a, b \in \mathbf{N} \}$

  Then the class $\mathsf{C} = \{ L(x^m y^n) \mid m, n \in \mathbf{N}\}$ is identifiable from positive representation.

Monomial and Hasse Diagram

$$x^m y^n \Rightarrow (m, n)$$