

Maschinelles Lernen

Andre Gass

Graph Based Induction

as a Unified Learning Framework

Nach KENICHI YOSHIDA, HIROSHI MOTODA
Und NITIN INDURKHYA

Ablauf

- Einführung
- Begriffe
- Graphen
- Umwandlung in Graphen
- CLiP
- Ergebnisse
- Fazit

Einführung

- Die Idee hinter GBI
 - Umformen von Problemen in Graphen
 - Allgemeine Betrachtung des Graphen
 - Modifikation des Graphen
- Vorteile dabei
 - Probleme können allgemein betrachtet werden
 - Keine spezielle Lösung mehr für verschiedene Lernprobleme
 - Ein Algorithmus für möglichst viele Aufgaben

Begriffe

- Gefärbter, gerichteter Graph
 - Graph (Knoten, Kanten)
 - Knoten sind gefärbt, eventuell mit mehreren Farben
- Muster
 - Häufig auftretende Teilgraphen
- Views
 - Mengen von Mustern

CLiP - Funktionsweise

- Es werden Muster im Graph gesucht
 - Muster sind Teilgraphen, die häufig auftreten
- Aus diesen Mustern werden Views generiert
- Views
 - werden erweitert
 - Es gibt nur eine begrenzte Anzahl

Finden von Teilgraphen

- Es wird nur nach identischen Mustern gesucht
- Isomorphe Teilgraphen werden nicht betrachtet
 - Spart Zeit und ist, nach den Autoren, ausreichend
 - Alle isomorphen Teilgraphen zu finden ist NP vollständig

Graphen

- Der Graph wird, unter Verwendung der Views verkleinert
 - Pattern werden zu Knoten zusammengefasst
 - Neue Knoten bekommen neue Farbe
- Ergebnis sind die Views
 - Bzw. Die enthaltenen Pattern
 - Diese müssen interpretiert werden

Finden von Teilgraphen

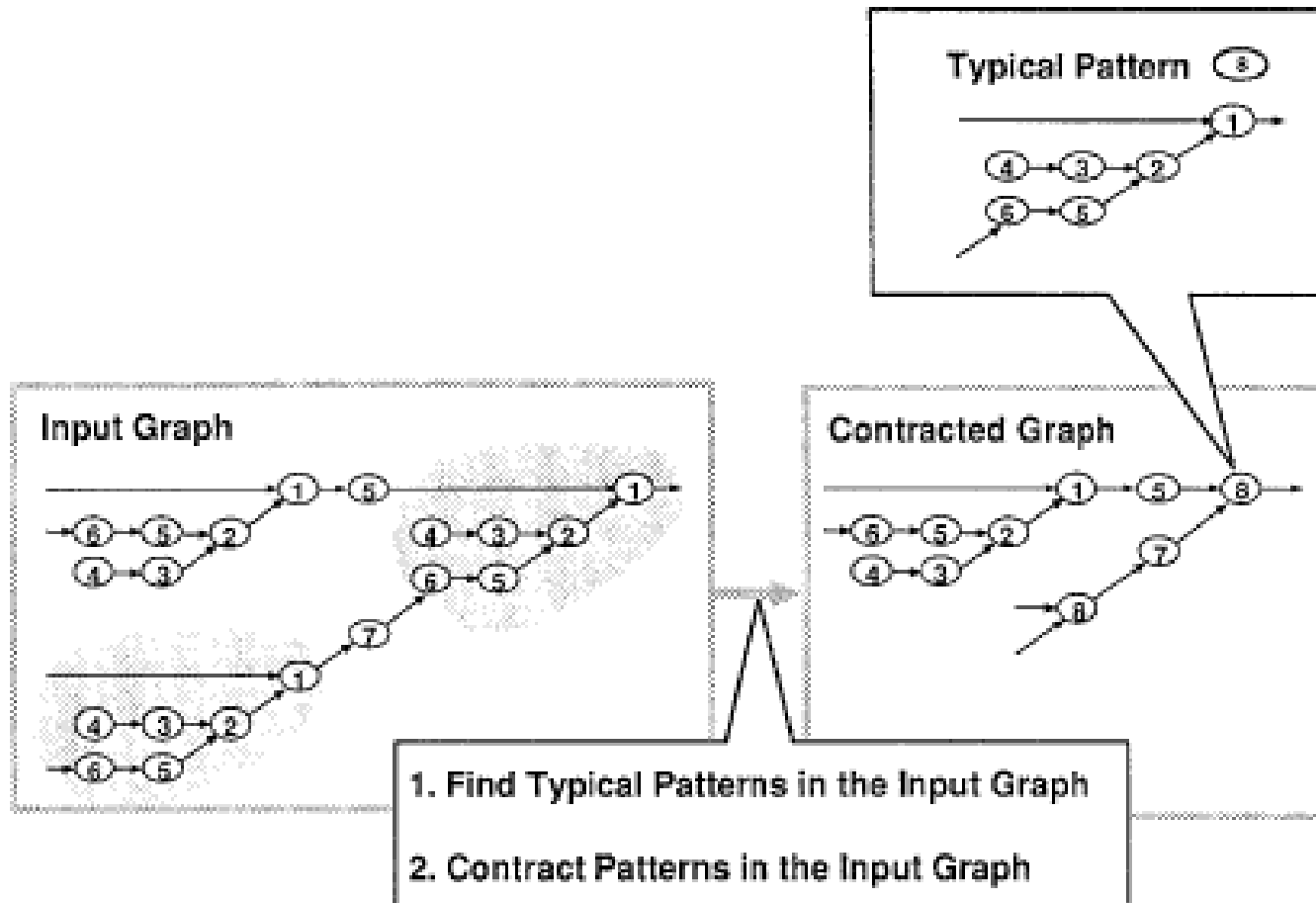


Fig. 1. Graph contraction by finding typical patterns.

Umwandlung in Graphen

- Klassifizierung von DNS-Sequenzen
- Makros um das Lösen von Gleichungen zu beschleunigen
- Finden von höheren Elementen in elektrischen Schaltungen

Graphen

- Klassifizierung von DNS-Sequenzen
 - In Promotor und nicht-Promotor

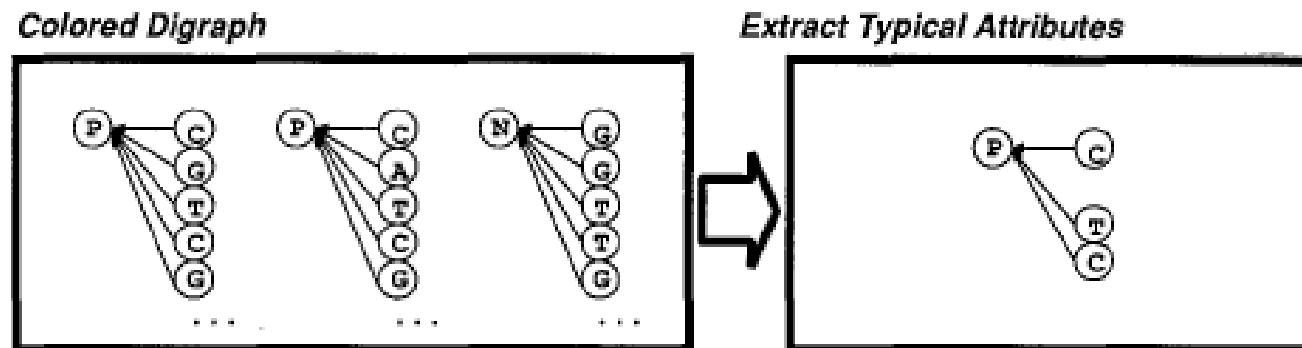


Fig. 2. Extraction of classification rules from DNA sequence data.

Graphen

- Makros

- Es wird nach neuen Regeln gesucht
 - Die mehrere alte Regeln zusammenfassen
- Zu einer schnelleren Abarbeitung der Probleme z.B. In Prolog führen

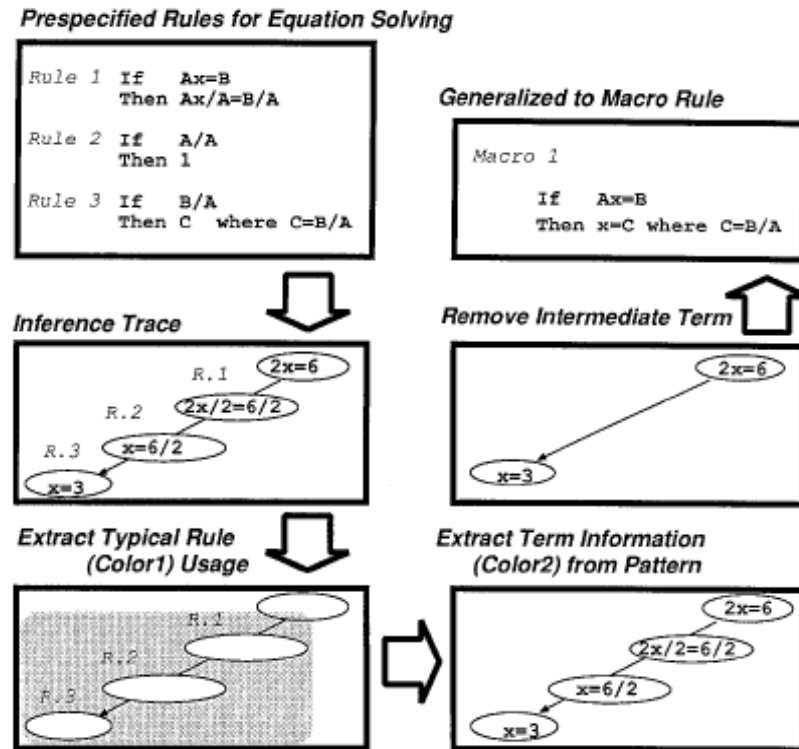


Fig. 4. Learning macro rules for equation solving.

Graphen

- Elektrische Schaltungen
 - Hierarchisches Modell
 - Konzepte auf höheren Ebenen sollen gefunden werden
 - Es soll problemlos möglich sein, auf die detailreichen Pläne der unteren Ebenen zurückzugreifen

CLiP

- Konstruktion von sog. Views
 - Bestehen aus Pattern, die häufig gefunden wurden
 - Bilden das Ergebnis des Algorithmus
- Pattern
 - Werden gesucht um die Views zu bilden
 - Und den Graphen zu verkleinern
 - Ergeben einen neuen Knoten im Graph
 - Der eine neue Farbe bekommt

CLiP

- Im nächsten Schritt werden die Pattern modifiziert/erweitert
 - Wieder wird gesucht, wie oft die Pattern vorkommen
 -
- Es ist wichtig, dabei eine sinnvolle Gewichtung zu verwenden
 - Üblicherweise wird dabei die Anzahl der Knoten gegen die Anzahl der neuen Farben abgewogen
 - Jedes Pattern muss mindestens zweimal vorkommen

CLiP

- Es gibt nur eine bestimmte Anzahl von Views
 - Die besten werden verwendet, der Rest verworfen
- Views enthalten immer auch die vorherigen Pattern
 - Diese werden nur durch die Neuen ergänzt

Pattern Modification

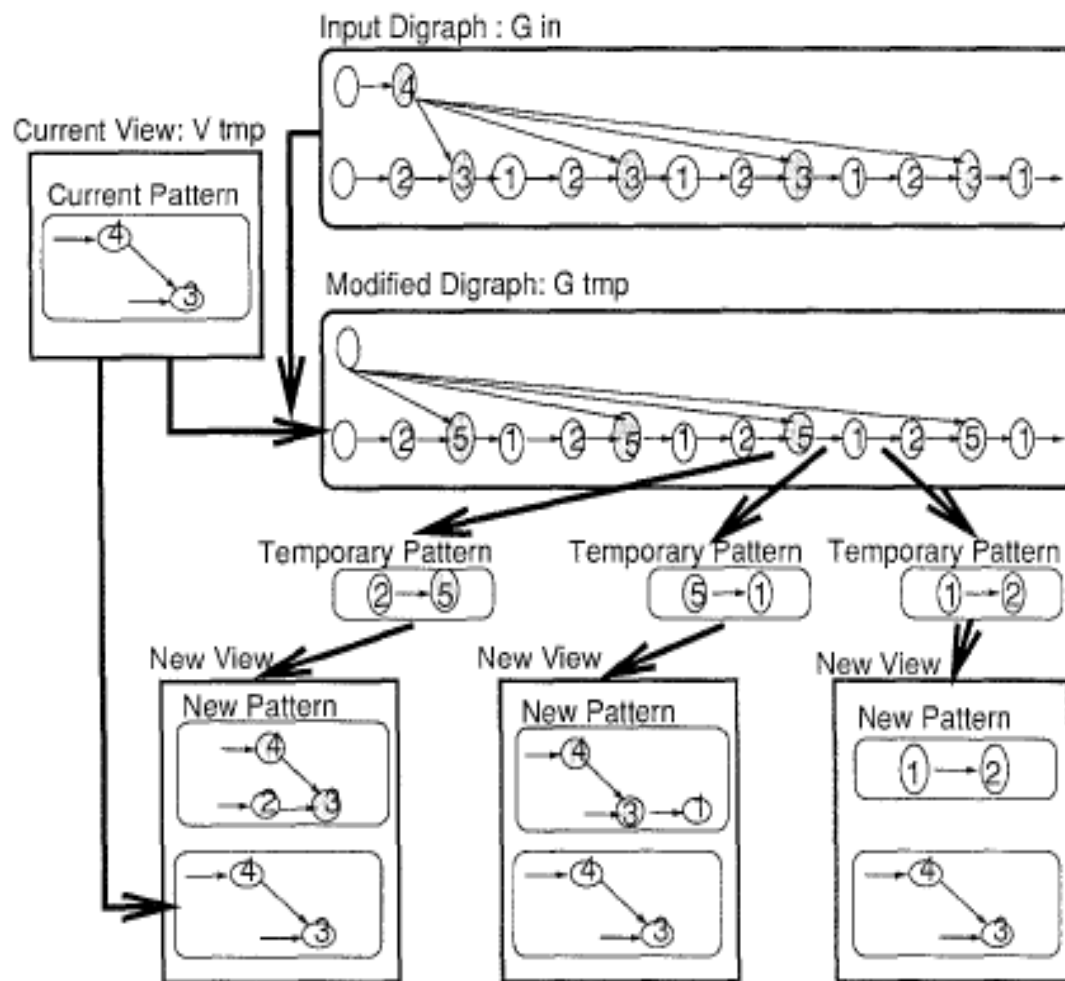


Fig. 10. Pattern modification.

CLiP

```
Algorithm CLiP( $G_{in}, C, L, W$ )
  Input       $G_{in}$       : Colored Directed Graph
              $C$          : Selection Criterion
              $L, W$       : integer
  Output     Sequence of  $V_i$  where each  $V_i$  is
             a Set of Typical Patterns in  $G_{in}$ 
  Variable    $B, B_{next}$  : Set of Views
begin
   $V_0 \leftarrow \emptyset$ ;    $B \leftarrow \{V_0\}$ ;    $i \leftarrow 1$ 
  repeat  $L$  do
     $B_{next} \leftarrow \emptyset$ 
    for each  $V_{tmp} \in B$  do
      Call Pattern Modification
      Call Pattern Combination
      Call View Selection
     $V_i \leftarrow$  Best view in  $B_{next}$  according to  $C$ 
     $i \leftarrow i + 1$ 
  return Sequence of  $V_i$ 
end
```

CLiP

Procedure *Pattern Modification*

begin

$G_{tmp} \leftarrow$ Graph that is contracted from G_{in}
according to the patterns in V_{tmp}

for each Temporary Pattern P in G_{tmp} **do**

$V_{new} \leftarrow V_{tmp} \cup \{\text{Original Pattern of } P\}$

Append V_{new} to B_{next}

end

Procedure *Pattern Combination*

begin

for each $V_{tmp1} \in B$ **do**

for each $V_{tmp2} \in B$ **do**

if $V_{tmp1} \neq V_{tmp2}$ **then**

$V_{new} \leftarrow V_{tmp1} \cup V_{tmp2}$

Append V_{new} to B_{next}

end

Procedure *View Selection*

begin

$B \leftarrow$ Top W views in B_{next} according to C

end

Fig. 9. Algorithm for extracting typical patterns.

Ergebnisse - Klassifizierung

- Finden von Promotoren in DNA-Sequenzen
- 106 DNS-Sequenzen
 - Bestehend aus jeweils 57 Nukleotiden
- Die Hälfte davon Promotoren

Table 1. Inductive learning: comparison with other classification methods

Method	Previously reported methods			
	ID3	SWAPI	BP	CLiP
Error/106	19	14	8	14

Ergebnisse - Makros

- Lösen von Gleichungen erster Ordnung
- Vorher 100 Sekunden
- Nachher 88 Sekunden CPU-Zeit

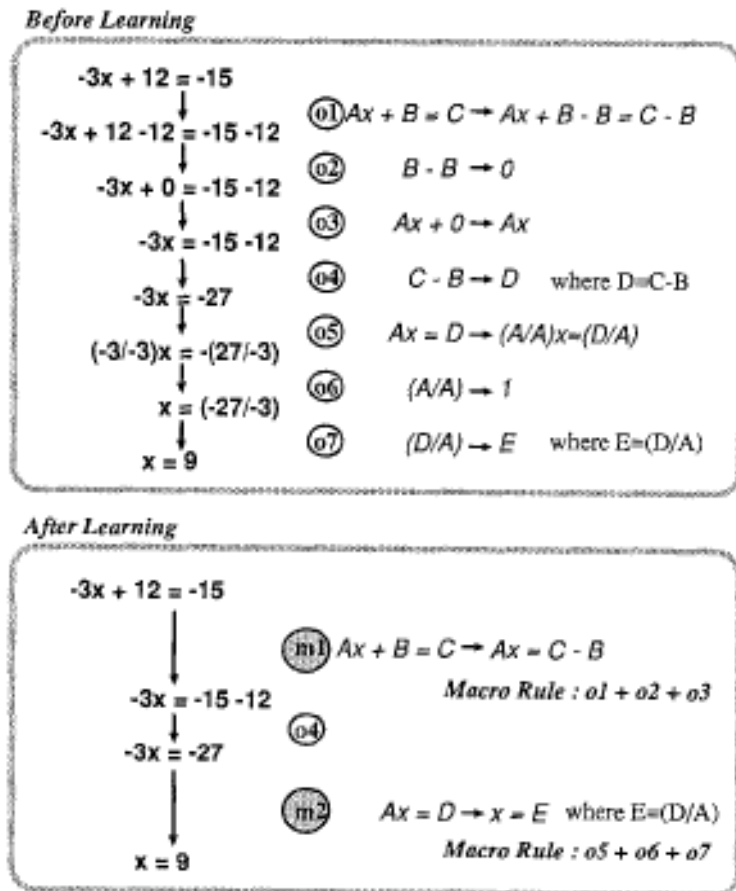


Fig. 12. Use of learned macro operators in equation solving.

Ergebnisse - Schaltungen

- Simulation einer NMOS-Schaltung
- Nach einer Anpassung der Gewichtung:

Table 3. List of generated concepts in the hierarchical knowledge base

No.	Generated Concept	Comment
1.	Pull Up Transistor	Circuit made up of pure transistor, capacitor, and power source
2.	Pull Down Transistor	Circuit made up of pure transistor, capacitor, and ground
3.	Analog NOT	Circuit made up of pull up transistor and pull down transistor Inference table contains analog element
4.	Analog NOR	Circuit made up of pull up transistor and 2 pull down transistors Inference table contains analog element
5.	Digital NOT	Similar to Analog NOT Inference table does not contain analog element
6.	Digital NOR	Similar to Analog NOR Inference table does not contain analog element
7.	Carry Chain	Circuit which calculates carry

Fazit

- GBI vereinfacht bestimmte Probleme erheblich
- Erzeugt in kurzer Zeit sinnvolle Ergebnisse
- Hat das Potential deutlich optimiert zu werden
 - Isomorphe Teilgraphen

Quellen

- Kenichi Yoshida, Hiroshi Motoda, Nitin Indurkha: Graph-based induction as a unified learning framework. Applied Intelligence 4(3): 297-316 (1994)