Master Recherche Orsay 2006-2007

Fouille de Données et Apprentissage

Michèle Sebag

TAO : Thème Apprentissage et Optimisation, Université Paris-Sud

http://tao.lri.fr/

Apprentissage relationnel

- Introduction
- Rappels de logique

  `http://www.cl.cam.ac.uk/Teaching/1998/LogProof`
- Programmation Logique Inductive (ILP)
- Une limite : la transition de phase
- Etude de cas : Apprentissage de lois et programmation génétique

Motivations

Représentations propositionnelles 90% des applications

Limites :

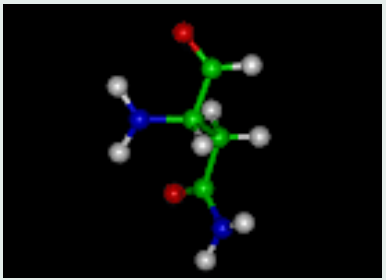
Quand un exemple est formé d'entités semblables en relation

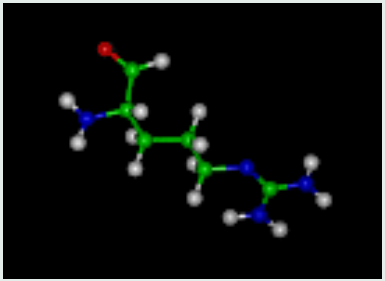
Exemples de domaines :

- Chimie (molécule : atomes en relation)
- Langage naturel (phrase : mots en relation)
- Urbanisme (paysage : batiments, routes, ...)

La frontière propositionnelle / relationnelle

Ne peut-on pas se ramener au problème précédent ?





Représenter la première molécule : facile

atom1	..	atomN	lien $1-2$	..	lien $i-j$
carbone	..	hydrogene	double	..	-

Représenter la seconde molécule : ...??

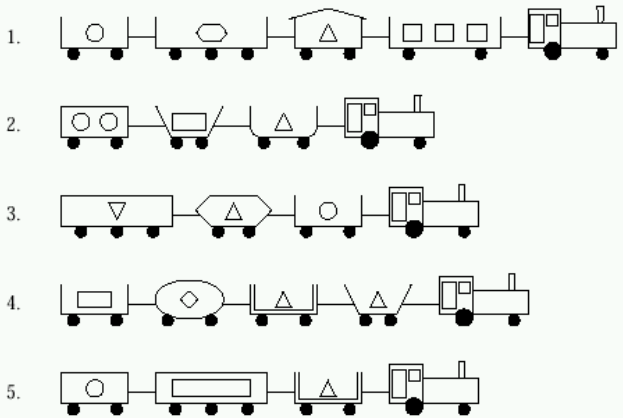
$\rightarrow$ Considérer tous les appariements :

atome i (premiere molécule) $\leftrightarrow$ atome j (seconde molécule)

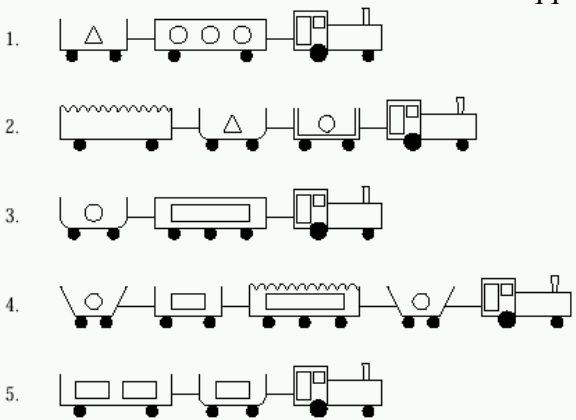
Un facteur exponentiel

Michalski's Trains

1. TRAINS GOING EAST



2. TRAINS GOING WEST



qq

Données structurées

Séquences — Bio-info, Fouille de textes, Gestion d'alarmes,..

Arbres — Données XML, fouille du Web,..

Graphes

Chimie — Predictive Toxicology Evaluation

$$muta(m) \leftarrow atm(m, m_1, Carb), \ldots atm(m, m_K, Hydr), bond(m_i, m_j, simple), \ldots, bond(m_p, m_q, double)$$

Fouille de données multi-relationnelles — CRM

$$bonClient(m) \leftarrow transaction(m, m_1), ..transaction(m, m_K) \; client(m), client(m_1), \ldots, client(m_K)$$

Apprentissage relationnel

- Introduction
- Rappels de logique

  `http://www.cl.cam.ac.uk/Teaching/1998/LogProof`
- Programmation Logique Inductive
- Apprentissage relationnel et transition de phase
- Etude de cas : Apprentissage de lois et programmation génétique

Logique

- Enoncés — *Black is the color of my true love's hair*
- Interprétation — vrai ou faux ?
- Logique — relations entre énoncés (consistance, implication)
- Preuve — modélisation du raisonnement humain déductif
- Apprentissage — raisonnement inductif

Logique du Premier Ordre

Décrire — Connaissances du domaine

$$
\begin{array}{lcl}
grandpere(X,Y) & \leftarrow & pere(X,Z), parent(Z,Y) \\
mutagen(m) & \leftarrow & atm(m,m_1,carbon, atm(m,m_2,hydr), .. \\
 & & bond(m_1,m_2,simple), ...
\end{array}
$$

Déduire

- Raisonner sur les fonctions et les relations entre entités

$$(pere(Jean, Sophie), pere(Sophie, Marc)) \quad \Rightarrow \quad grandpere(Jean, Marc)$$

- Raisonner sur les rapports entre tous et quelques uns

$$(homme(X) \Rightarrow mortel(X)), homme(Socrate) \quad \Rightarrow \quad mortel(Socrate)$$

Logiques

Niveaux

- Logique propositionnelle : logique booleenne classique
- Logique relationnelle du premier ordre :
  variables, quantification universelle ou existentielle
- Logiques d'ordres supérieurs :
  raisonnement sur les ensembles et les fonctions
  (applications à la vérification de hardware)
- Logique modale : raisonner sur ce qui doit, ou peut, arriver

Logique du premier ordre, 2

- Fonction — f arité n
- Constante (arité 0) — $m_1, carbon$
- Variable — X, Y, Z
- Prédicat, fonction à valeurs dans $\{V, F\}$ — $grandpere, atm, bond$
- Terme t — Variable, constante ou $f(t_1, \ldots, t_n)$
- Atome — $pere(X, Z), atm(m, m_1, carbon)$
- Littéral — Atome ou négation d'un atome
- Clause — $L_1 \vee \ldots \vee L_n$
- Clause définie : un seul littéral négatif

représenter $\equiv$ calculer

Connaissance du domaine

$$parent(X, Y) \leftarrow mere(X, Y)$$
$$parent(X, Y) \leftarrow pere(X, Y)$$
$$grandparent(X, Y) \leftarrow parent(X, Z), parent(Z, Y)$$

Faits — Interprétation

$$pere(Jean, Sophie), pere(Sophie, Marc), \neg mere(Marc, Jean)...$$

Propriétés d'un énoncé — fonction des Interprétations qui le satisfont

- Valide — toutes
- Contingent — certaines
- Insatisfiable — aucune

représenter ≡ calculer, 2

Implication logique $\models$

Toute interpretation qui satisfait les prémisses satisfait la conclusion

$$\{p\} \models (p \vee q)$$
$$\{p\} \not\models (p \wedge q)$$
$$\{p, q\} \models (p \wedge q)$$

Règles d'inférence

Modus Ponens

$$\frac{\phi \Rightarrow \psi \quad \phi}{\psi}$$

Modus Tollens

$$\frac{\phi \Rightarrow \psi \quad \neg\psi}{\neg\phi}$$

Double négation

$$\frac{\neg\neg\phi}{\phi}$$

Elimination d'equivalence

$$\frac{\phi \Leftrightarrow \psi \quad \phi \Rightarrow \psi}{\psi \Rightarrow \phi}$$

Règles d'inférence, suite

Règle de résolution

Principe : $(A \vee B) \wedge (\neg A \vee C) \models (B \vee C)$

Ma montre est arrêtée ou cet homme est mort
Cet homme est vivant ou je suis Groucho Marx

Ma montre est arrêtée ou je suis Groucho Marx

$$\frac{\{A, B_1, \ldots, B_n\} \qquad \{\neg A, C_1, \ldots, C_m\}}{\{B_1, \ldots, B_n, C_1, \ldots, C_m\}}$$

Cas particuliers

Réduction

$$\frac{\{A\} \qquad \{\neg A, B_1, \ldots, B_n\}}{\{B_1, \ldots, B_n\}}$$

Incohérence

$$\frac{\{A\} \qquad \{\neg A\}}{\square}$$

Logique du premier ordre

Variables liées et libres

$$\forall X \exists Y\, p(X, Y), q(Y, Z)$$

Substitution

Un ensemble fini de remplacement variable/terme

$$\begin{array}{ll} & \forall X\, homme(X) \Rightarrow mortel(X) \\ (X/Socrate) & homme(Socrate) \Rightarrow mortel(Socrate) \\ (X/table) & homme(table) \Rightarrow mortel(table) \end{array}$$

Unification

θ unifie t et t' si $t\theta = t'\theta$

θ est plus général que θ' si $\theta' = \theta o \sigma$

mgu(t, t') : substitution max. general unifiant t et t'

$$mgu(t = f(a, X), t' = f(Y, g(Z))) : \theta = \{Y/a, X/g(Z)\}$$

Substitutions

Clause

C $mutagen(X) \leftarrow atm(X,Y,carbon), atm(X,Z,carbon), atm(X,T,hydr)$
$bond(Z,T,simple)$

E $mutagen(m) \leftarrow atm(m,m_1,carbon), atm(m,m_2,carbon), atm(m,m_3,carbon)$
$atm(m,m_4,hydr), bond(m_1,m_2,simple), bond(m_3,m_4,simple)$

Substitution

$$\theta = \{X/m,\ Z/m_3,\ T/m_4\} \qquad C\theta \subset E$$

θ-subsumption

Plotkin

$$C\theta \subset E : C \models E$$

Rq: plus faible que l'implication logique

Démonstration – Preuve

Etant donné $\{$ expressions $\}$, prouver (expression but)

Comment faire

Séquence $S_1, \ldots, S_K$ d'application de règles d'inférence telle que

Etape i :

Prémisses : expressions initiales ou
résultats obtenus aux étapes $1 \ldots i-1$

Etape K :

Résultat : (expression but)

Apprentissage relationnel

- Introduction
- Rappels de logique

  `http://www.cl.cam.ac.uk/Teaching/1998/LogProof`
- Programmation Logique Inductive
- Apprentissage relationnel et transition de phase
- Etude de cas : Apprentissage de lois et programmation génétique

ILP : Problème posé

Input
- Exemples $\mathcal{E}^+, \mathcal{E}^-$
- Théorie du domaine $\mathcal{B}$
- Espace d'hypothèses $\mathcal{H}$
- Relation de couverture, implication logique $\models$

Propriétés cherchées
- Complétude $\mathcal{B}, h \models e,\ e \in \mathcal{E}^+$
- Correction $\mathcal{B}, h \not\models e,\ e \in \mathcal{E}^-$

Formulation faible
- Implication logique indécidable
- Theta-subsumption Plotkin, 70

Programmation Logique Inductive

Les fondamentaux

- Tout est dans la représentation : pas d'appauvrissement !

  molécule $\notin \mathbb{R}^d$, (masse, charge, hydrophobicité,..)

- Utilisation rigoureuse de la connaissance du domaine

  déclaratif vs procédural

- Que l'induction soit un mode de programmation

  Le Graal : la synthèse de programme

Même esprit que Prolog

non pas que le programmeur peine à décrire le COMMENT pour que la machine l'exécute aisément

mais que le programmeur décrive aisément le QUOI et que la machine se débrouille pour passer du QUOI au COMMENT

Programmation Logique Inductive, 2

Première époque 1990-1997

- Synthèse automatique de programmes à partir de traces

$$sort(L) \leftarrow list(L, [X, L']), list(L', [Y, L'']), sort([X, Y]), sort(L'')$$

Cœur algorithmique :

- Espace de recherche : programmes Prolog
- Approches de type Generate & Test
- Critères d'optimisation, Heuristiques d'élagage

Priorités

- Apprendre avec peu d'exemples les bons
- Trouver "la" solution
- Récursivité

PLI, Seconde époque (1995-...)

Predictive Toxicology Evaluation

- Muggleton-King-Srinivasan, 96-06
- De Raedt-Kramer, 01-05

Scientific Discovery

- Identification of developmental laws, Dzeroski et al. 97-..
- Identification of behavioral laws, Sebag et al. 96-02

Priorités

- Résister au bruit des données
- Traiter les informations numériques
- Efficacité algorithmique

Programmation Logique Inductive
deux formulations

Learning from Interpretation

Tables relationnelles

- pere(Jean,Sophie), mere(Sophie,Marc),..
- grandpere(Jean,Marc)

Learning from Entailment

- grandpere(Jean,Marc) $\leftarrow$ pere(Jean,Sophie),pere(Sophie,Marc)

ILP, Algorithmes

Alg. descendants — top-down

- Approche générer et tester
- Init : une clause très générale — tc(X) ←
- Iterativement, spécialiser — appliquer des substitutions ou ajouter des littéraux dans le corps de la clause

Alg. ascendants — bottom-up

- Une approche guidée par les données
- Init : une clause très spécifique — (l'exemple; ou "bottom clause")
- Itérativement, généraliser — appliquer des substitutions inverses ou enlever des littéraux

FOIL

Quinlan 90

- Init :
  - $H = \{\}$
  - $\mathcal{E}$ = {ensemble des exemples positifs}
- Tant que $\mathcal{E}$ n'est pas vide
  - $C : tc(X) \leftarrow Body$, $Body = true$
  - Jusqu'à ce que C soit correct
    - Spécialiser C
  - $H = H \cup \{C\}$
  - Oter de $\mathcal{E}$ les exemples couverts par C

FOIL, suite

Spécialiser $C : tc(X) \leftarrow Body$

- Pour $L \in \rho(Body)$ spécialisations de $Body$
  - Calculer $critere(L)$
- $Body \leftarrow Body \wedge argmax\{critere(L), L \in \rho(Body)\}$

Critère $c(Body \wedge L)$

- pureté $Pr(+|Body \wedge L)$
- quantité d'information - pureté log (pureté)
- gain pureté $(Body \wedge L)$ - pureté $(Body)$

Relationnel, ce qui change

Clauses connectées — partagent des variables

- $L \in \rho(Body)$ si $Body \wedge L$ est connecté

Look ahead — la myopie est plus grave

Ex :

$$fume(X) \leftarrow ami(X,Y), fume(Y)$$

Il faut considérer $L = ami(X,Y)$, alors qu'il n'apporte aucune information, si on veut pouvoir accéder à $fume(Y)$

Evaluation plus complexe

$n^-(Body)$: nombre de substitutions θ tq $Body\theta$ couvre un négatif

prendre en compte

$$\frac{n^-(Body \wedge L)}{n^-(Body)}$$

PROGOL

Muggleton 95

Inversion de la résolution

Principe

- B : connaissance du domaine
- E : exemples
- Trouver H tq $B, H \models E$

$$B \not\models E$$

$$\Rightarrow B, \neg E \models \neg H$$

Algorithme

- Construire la bottom clause $\mathcal{B} = B \wedge \neg E$
- se ramener au problème précédent : ne considérer que les spécialisations de $Body$ qui généralisent $\mathcal{B}$

ILP Applications

Ecology

- Biological classification of river water quality
- Modelling algal growth in the Lagoon of Venice
- Modelling growth of maximal biomass quantity in the metalimnion of the east basin of the lake of Bled
- Predicting biodegradability of chemical compounds

Discussion

- \- Handling numerical knowledge
- \+ Exploit background knowledge

ILP Applications, biologie moléculaire

La mutagenèse

- 230 molecules; circa 40 atoms, 60 bonds.
- Effets de la connaissance du domaine
  - atomes et liens
  - idem + “connaissances” numériques
  - idem + connaissances chimiques (e.g. groupes méthyl)

ILP Applications, biologie moléculaire, 2

Résonance magnétique nucléaire, spectre de fréquences

Les diterpènes

- 1503 examples (diterpene molecules)
- 23 target relations (labdan(Molecule), clerodan(Molecule), ...)
- background knowledge

Comparaisons (10cv)

	FOIL	RIBL	C4.5
red	46.5	86.5	—
prop	70.1	79	78.5
red + prop	78.3	91.2	—

Fouille de données relationnelles

Diagnosis-Therapy Index

- Data gathered from a survey of German hospitals.
- multirelational database with 6 relations:
  - hospitals and units [300 tuples]
  - patients [6.000 tuples]
  - diagnoses [25.000 tuples]
  - therapies [43.000 tuples]
  - patient-therapies-days [260.000 tuples]
  - patient-diagnosis-days [250.000 tuples]
- Goal: find groups of patients (therapies, hospitals) with unusual cost or success structure

Diagnosis-Therapy Index

Representation

- patient(PatientID,Name,Age,Sex,Outcome,...)
- patient_diagnosis(PatientID,DiagnosisID,Date,HospitalID)
- patient_therapy(PatientID,TherapyID,Dosage,Date,HospitalID)
- diagnosis(DiagnosisID,Name,Latin)
- therapy(TherapyID,Name,Duartion,StandardMedification)
- hospital(HospitalID,Name,Location,Size,Owner,Class)

"Patients older than 65 who were diagnosed at a small hospital have an unusally high mortality rate."
$patient(I, N, A, S, O), (A > 65), p_d(I, D, Dt, H), hospital(H, *, *, s$

Fouille de données relationnelles, 2

quand les données ne tiennent pas en mémoire

Exploration

- Langage : e.g., clauses connectées; suivre les clés
- Target attribute: treatment success (binary, yes or no)
- Reference population: distribution [61%, 31%]
- "Patients older than 65 who were first diagnosed in a small hospital"

  distribution [43%, 57%]

Apprentissage relationnel

- Introduction
- Rappels de logique

  `http://www.cl.cam.ac.uk/Teaching/1998/LogProof`
- Programmation Logique Inductive
- Apprentissage relationnel et transition de phase
- Etude de cas : Apprentissage de lois et programmation génétique

Apprentissage relationnel

Les questions qui fâchent :

- Passage à l'échelle ?
- *Where are the really hard problems* ?

Problème posé

Input
- Exemples $\mathcal{E}^+, \mathcal{E}^-$
- Théorie du domaine $\mathcal{B}$
- Espace d'hypothèses $\mathcal{H}$
- Relation de couverture, implication logique $\models$

Propriétés cherchées
- Complétude $\mathcal{B}, h \models e,\ e \in \mathcal{E}^+$
- Correction $\mathcal{B}, h \not\models e,\ e \in \mathcal{E}^-$

Formulation faible
- Implication logique indécidable
- Theta-subsumption Plotkin, 70

Theta-subsumption $\equiv$ Constraint Satisfaction Problem

Θ-subsumption : $C \prec D$ *iff* $\exists \theta / C\theta \subseteq D$

$$
\begin{array}{lccc}
C: & q_0(X_1, X_2), & q_1(X_1, X_3), & q_2(X_2, X_3) \\
D: & q_0(a_2, a_7), & q_1(a_3, a_2), & q_2(a_1, a_4), \\
 & q_0(a_3, a_1), & q_1(a_7, a_2), & q_2(a_7, a_2), \\
 & .. & .. & ..
\end{array}
$$

Constraint Satisfaction : trouver θ

Contraintes $\boxed{q_0(X_1, X_2) \wedge q_1(X_1, X_3) \wedge q_2(X_2, X_3)}$

Relation q_0 : $\{q_0(a_1, a_2),\ q_0(a_2, a_7), ..\}$

Substitution θ : $\{X_1, .., X_n\} \rightarrow \{a_1, .., a_L\}$

Solution θ: $q_0(\theta(X_1), \theta(X_2)) \in$ Relation q_0

$q_1(\theta(X_1), \theta(X_3)) \in$ Relation q_1

$q_2(\theta(X_2), \theta(X_3)) \in$ Relation q_2

Θ-subsumption

$$C: \quad p_1(X_1, X_4), p_2(X_1, X_2), p_3(X_2, X_3), \ldots\ldots p_m(X_2, X_4)$$

$$\theta = (X_1/a_{17}, X_2/a_5, X_3/a_9, X_4/a_{11})$$

D

p_1	p_2	p_3	$\ldots$	$\ldots$	p_m
a_8, a_{15}	a_{21}, a_{10}	a_5, a_9			a_{14}, a_4
a_{17}, a_{11}	a_3, a_{16}	a_{18}, a_{19}			a_7, a_{15}
a_{20}, a_{12}	a_{17}, a_5	a_{14}, a_{11}			a_5, a_{11}
$\ldots$	$\ldots$	$\ldots$			$\ldots$

Transition de phase et CSP

Complexité pire cas : exponentielle $\quad |\text{search space}| = L^n$

souvent $>>$ complexité effective

Paramètres d'ordre CSP

$p_1 = \frac{2m}{n(n-1)}$ densité de contraintes

$p_2 = 1 - \frac{N}{L^2}$ dureté des contraintes

Transition de phase en CSP $\quad$ Modèle statistique

Soit $csp(p_1, p_2)$ une instance de CSP de paramètres p_1, p_2

Satisfiabilité moyenne

$$P_{sol}(p_1, p_2) = Pr(csp(p_1, p_2) \text{ satisfiable})$$

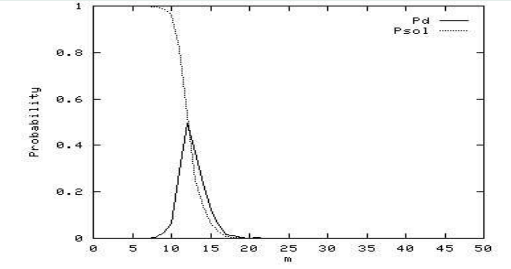
Coût moyen

$$Compl(p_1, p_2) = \mathbb{E}[Complexite(csp(p_1, p_2))]$$

Transition de phase en CSP

Expérimentalement: Fixer p_1, varier p_2 de 0 à 1.

p_2	P_{sol}	Cot	Region
petit	≈ 1	-	YES
—	$1 \searrow 0$	elevé	PT
grand	≈ 0	-	NO



Transition de phase : Lieu des problèmes difficiles en moyenne.

Θ-subsumption : Existence de TP

Giordana Saitta, MLJ 00

Paramètres d'ordre

n	nb variables	N	nb littéraux
m	nb prédicats	L	nb constantes

Protocole

Pour $n \in [4, 14],\ m \in [5, 50], N \in [50, 100],\ L \in [15, 50]$

- Tirer 100 paires (C, e)

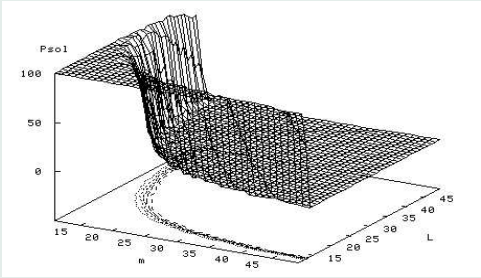
$$C = p_1(x_{1,1}, x_{1,2}), \ldots, p_m(x_{m,1}, x_{m,2})$$

$$e = \bigwedge_{k=1}^{m} p_k(a_{1,1}, a_{1,2}), \ldots, p_k(a_{N,1}, a_{N,2})$$

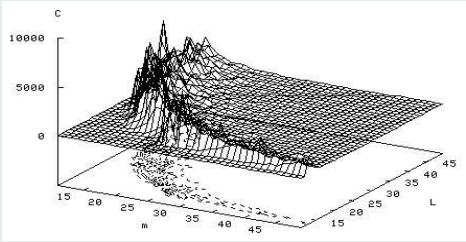
- Mesurer $P_{sol}(n, m, N, L) = Pr(C \prec e)$

Pour $n = 10, N = 100$

Probabilité de couverture(m, L)



Complexité effective(m, L)



Observations

Comme attendu :

- Existence d'une région OUI

  C trop générale wrt e, $Pr(C \prec e) \approx 1$

- Existence d'une région NON

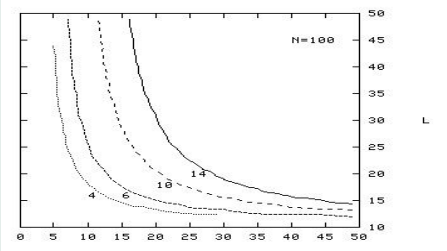
  C trop spécifique wrt e, $Pr(C \prec e) \approx 0$

- Existence d'une étroite transition de phase,

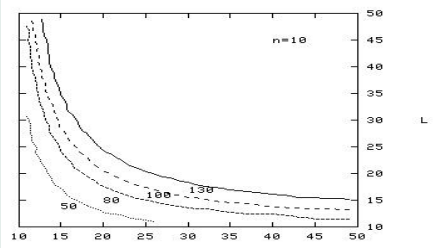
  où le coût de la θ-subsumption est maximum.

Lieu de la transition de phase, $P_{sol} = .5$

$n = 4, 6, 10, 14$



N = 50,80,100



Conséquences sur l'apprentissage relationnel

Botta et al, 03

Protocole

1. Problèmes artificiels $(C, \mathcal{E})$ — Régions : OUI, NON, TP
2. Apprendre $\hat{C}$ — Algs. FOIL, Smart+, GNet
3. Etude : impact de la position du problème
   - sur la prédiction — $Err(\hat{C})$
   - sur la découverte — $C \neq \hat{C}$
   - sur le coût

Protocole expérimental

$n = 4$	nb variables	$N = 100$	nb littéraux
m	nb prédicats	L	nb constantes

Pour $m \in [5, 50]$, $L \in [15, 50]$

• Construire $\mathcal{C}$ $\quad x_{i,j} \in \{x_1, ..x_4\}$

$$p_1(x_{1,1}, x_{1,2}), \ldots, p_m(x_{m,1}, x_{m,2})$$

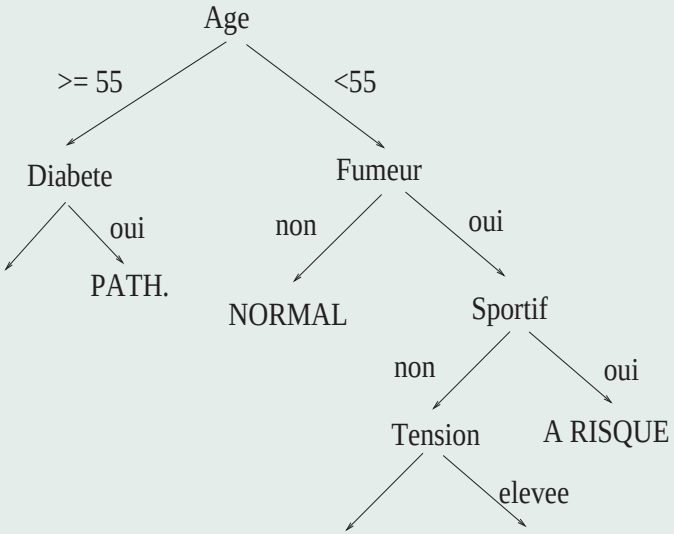
• Construire base d'apprentissage $\mathcal{E}_L$ et base de test $\mathcal{E}_T$:
200 exemples chaque, 100 positifs, 100 négatifs (Réparer e si nécessaire)

$$e = \bigwedge_{k=1}^{m} p_k(a_{1,1}, a_{1,2}), \ldots, p_k(a_{N,1}, a_{N,2})$$

• $\hat{C}$ = FOIL $(\mathcal{E}_L)$: succès ssi

$$|\{e \in \mathcal{E}_T \ / \ (C \prec e) \neq (\hat{C} \prec e)\}| < 20\%|\mathcal{E}_T|$$

FOIL : First Order Inductive Learner
1. Arbre de décision



Decision tree

Breiman et al. 83, Quinlan 79

Decision Tree $X = \mathbb{R}^d$

- Init: $T = \perp,\ node = \perp, E = \mathcal{E}, A = \{1..d\}$
- Recursively, find att_i

$$att_i = Argmax\{\text{Information Gain}(att_j), j \in A\}$$

$$\begin{array}{rrcl} & IG(att_i) & = & \sum_{v_j} Pr(att_i = v_j) IG(att_i = v_j), \\ \text{with} & IG(att_i = v_j) & = & -p_{i,j} log(p_{i,j}) - (1 - p_{i,j}) log(1 - p_{i,j}), \\ & p_{i,j} & = & Pr(y = +1 | x_i = v_j) \end{array}$$

- call Decision Tree

  $T \leftarrow (T \cup (edge(node, [att_i = v_j]))$,
  $node \leftarrow [att_i = v_j]$,
  $E \leftarrow E_{att_i = v_j}$,
  $A \leftarrow A - \{i\}$

FOIL : First Order Inductive Learner, 2

Quinlan 86

First step : propositional

- Find $f(X_1, X_2)$, $f = Argmax\{IG(q_j), j = 1..m\}$

Following steps

Given $f(X_1, ..X_k)$, find $q_p(X_i, X_j)$ st

- $f(X_1, ..X_k) \wedge q_p(X_i, X_j)$ connected
- $q_p(X_i, x_j) = ArgMax\{IG(f(X_1, ..X_k) \wedge q(X_t, X_u))\}$

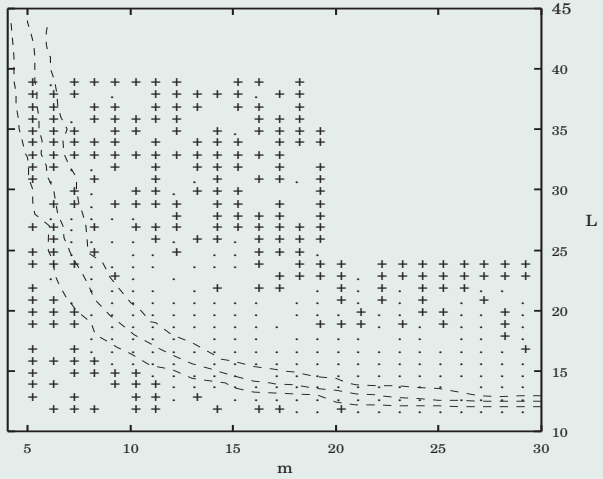
where

$$IG(f \wedge q(X_t, X_u)) = \mathcal{F}(\Theta^+(f), \Theta^+(f \wedge q), \Theta^-(f), \Theta^-(f \wedge q))$$

$$\Theta^+(g) = \{\theta, g\theta \subseteq e,\ e \in \mathcal{E}^+\}$$

$$\Theta^-(g) = \{\theta, g\theta \subseteq e,\ e \in \mathcal{E}^-\}$$

FOIL Competence Map



\+ Success ($> 80\%$ on test set) · Failure

$n = 4, N = 100$

Analyse de la carte de compétence

$\mathcal{C}$		$\hat{\mathcal{C}}$		Performances			Qualité		
m	L	K	$\hat{m}$	*train.*	*test*	CPU	Exact	%	
8	16	1	8	100	100	106.2	O	O	**Facile**
10	13	1	14	100	99	144.2	O	O	**F**
10	16	8	11.75	88	48.5	783.5	N	N	**Dur**
11	13	1	11	100	100	92.2	O	O	**F**
11	15	6	13.5	85	53.5	986.2	N	N	**D**
12	13	3	14	98.5	83	516.4	N	O	**faisable**
$\mathcal{C}$ appartient à la région p.s. satisfiable									
15	29	1	6	100	100	185.3	N	O	**f**
15	35	2	6	97.5	84.5	894.6	N	O	**f**
18	35	1	6	100	100	201.0	N	O	**f**
21	18	8	4.13	81.5	58	1394.9	N	N	**D**
25	24	1	6	100	99	135.9	N	O	**f**
29	17	1	12	100	99.5	144.9	N	O	**f**
$\mathcal{C}$ appartient à la région p.s. insatisfiable									
6	28	12	8.08	91.5	50.5	815.4	N	N	**D**
7	28	11	7.63	91.5	60.5	1034.2	N	N	**D**
8	27	1	7	100	100	58.8	O	O	**F**
13	26	1	9	100	99	476.8	N	O	**f**
17	14	8	15	93	46	294.6	N	N	**D**
18	16	8	8.87	91	58.5	404.0	N	N	**D**
26	12	3	24.33	80	58	361.4	N	N	**D**
$\mathcal{C}$ appartient à la transition de phase									

1. Problèmes faciles

$\mathcal{C}$		$\widetilde{\mathcal{C}}$		Performances			Qualité	
m	L	K	$\hat{m}$	*train.*	*test*	CPU	Exact	%
8	16	1	8	100	100	106.2	O	O
10	13	1	14	100	99	144.2	O	O
11	13	1	11	100	100	92.2	O	O

$\mathcal{C}$ appartient à la région p.s. satisfiable

8	27	1	7	100	100	58.8	O	O

$\mathcal{C}$ appartient à la transition de phase

Problèmes faciles

- Région satisfiable
- Concept cible petit

2. Problèmes faisables

$\mathcal{C}$		$\bar{\mathcal{C}}$		Performances			Qualité	
m	L	K	$\hat{m}$	*train.*	*test*	CPU	Exact	%
12	13	3	14	98.5	83	516.4	N	O

$\mathcal{C}$ appartient à la région p.s. satisfiable

15	29	1	6	100	100	185.3	N	O
15	35	2	6	97.5	84.5	894.6	N	O
18	35	1	6	100	100	201.0	N	O
25	24	1	6	100	99	135.9	N	O
29	17	1	12	100	99.5	144.9	N	O

$\mathcal{C}$ appartient à la région p.s. insatisfiable

13	26	1	9	100	99	476.8	N	O

$\mathcal{C}$ appartient à la transition de phase

Problèmes faisables

- Du bon côté de la TP
- Dans la région insatisfiable, très loin de la TP.

3. Problèmes difficiles

$\mathcal{C}$		$\bar{\mathcal{C}}$		Performances			Qualité		
m	L	K	$\hat{m}$	*train.*	*test*	CPU	Exact	%	
10	16	8	11.75	88	48.5	783.5	N	N	**D**
11	15	6	13.5	85	53.5	986.2	N	N	**D**

$\mathcal{C}$ appartient à la région p.s. satisfiable

21	18	8	4.13	81.5	58	1394.9	N	N	**D**

$\mathcal{C}$ appartient à la région p.s. insatisfiable

6	28	12	8.08	91.5	50.5	815.4	N	N	**D**
7	28	11	7.63	91.5	60.5	1034.2	N	N	**D**
17	14	8	15	93	46	294.6	N	N	**D**
18	16	8	8.87	91	58.5	404.0	N	N	**D**
26	12	3	24.33	80	58	361.4	N	N	**D**

$\mathcal{C}$ appartient à la transition de phase

Problèmes durs

- Grands concepts cibles
- Proches de la TP.

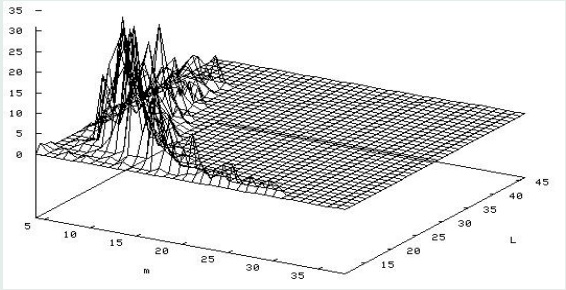
1. La TP est un attracteur de la PLI

Les hypothèses retenues sont dans la TP

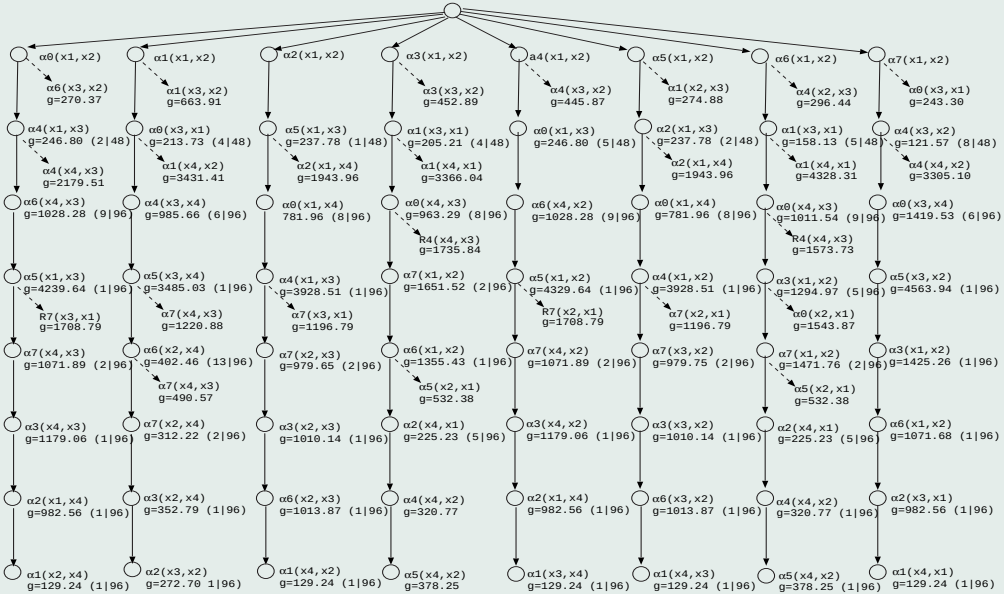
que le concept cible y soit ou non.

A posteriori, pas étonnant.

...Jette un doute sur le passage à l'échelle de la PLI...



2. La zone aveugle de la PLI



Pourquoi plus complexe $\not\Rightarrow$ plus difficile ?

I. Bonnes approximations

Soit C ds la région non satisfiable,
Soit $G \prec C$

G complet par construction

G appartient à la TP (versant insatisfiable), G p.s. correct
$\Rightarrow G$ bonne approximation

II. Probabilité de les trouver

$\#\{G \,/\, G \prec C,\ G \in TP\}$ exponentiel en la taille de C.

Apprentissage relationnel

- Introduction
- Rappels de logique

  `http://www.cl.cam.ac.uk/Teaching/1998/LogProof`
- Programmation Logique Inductive
- Apprentissage relationnel et transition de phase
- Etude de cas : Apprentissage de lois et programmation génétique

Exemple

	$Battery$	$Wire$	I	C	I/C
e_1	A	X	3.4763	3.4763	1.0000
e_2	A	Y	4.8763	4.8763	1.0000
e_3	A	Z	3.0590	3.0590	1.0000
e_4	B	X	3.9781	3.4763	1.1444
e_5	B	Y	5.5803	4.8763	1.1444
e_6	B	Z	3.5007	3.0590	1.1444
e_7	C	X	5.5629	3.4763	1.6003
e_8	C	Y	7.8034	4.8763	1.6003
e_9	C	Z	4.8952	3.0590	1.6003

But : Inférer des lois physiques (chimiques,..)

$$U = RI$$

Identification of macro-mechanical models

coll. M. Schoenauer (CMAP), and H. Maitournan (LMS)

Behavioral law of materials

- needed for accurate CAD;
- ill-known for new materials (e.g. polymers).

Art of macro-mechanical modeling:

- Adapting the model of another material;
- Designing a brand new model;
- Starting with a micro-mechanical analysis.

Fails when the current material:

- does not resemble other materials;
- does not fit expert's guesses;
- is not provided a tractable model by μ-M analysis.

Machine Discovery

- **First Era (1983)**

  Langley, Falkenhainer, Nordhausen,...

  Heavy assumptions
  Heuristic construction of new terms

  PV, PV/T, PV/nT,..

- **Second Era (1995)**

  Dzeroski, Valdes-Perez,...

  Exhaustive exploration
  Challenge : restricting the search space

- **On the ILP side**

  Srinivasan, Camacho, Simon, Frisch,...

  Fixed equations
  Parametric optimization

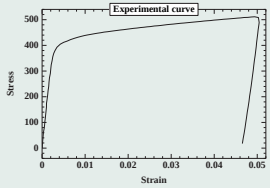
Dilemma

- **Strong background knowledge**
  $\longrightarrow$ exhaustive exploration is feasible

- **Reasonable background knowledge**
  + smart optimization criterion
  = greedy search

- **"Light" background knowledge**
  + stochastic search
  = global optimization

Identification of Behavioral Laws



Input: Experimental curves

- observed strain $\epsilon(t)$ for applied stress $\sigma(t)$;
- observed stress $\sigma(t)$ for applied strain $\epsilon(t)$;

Identification of Behavioral Laws, 2

Output: Behavioral law

Differential equations linking $\epsilon(t)$, $\sigma(t)$ and their derivatives, e.g.

if	$\sigma(t) < \sigma_1$	then $\sigma(t) = a.\epsilon(t) + b.\dot{\epsilon}(t)$
else if	$\sigma(t) < \sigma_2$	then $\sigma(t) = c.\epsilon(t) + d.\dot{\epsilon}(t)$

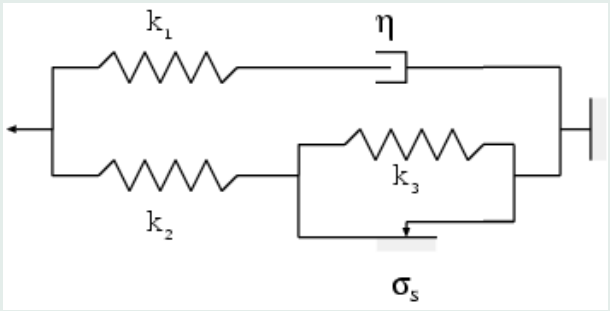
Criteria: the law must fit the experiments **and** be comprehensible.

Search space: Rheological models

Dynamic 1-D laws.

Assembly in series or parallel of

- springs (elastic behavior)
- sliders (plastic behavior)
- dashpots (viscous behavior)



Identification Goals:

- For a given model, adjust the parameters
  $\Longrightarrow$ **Parametric optimization**
- Optimize both the model and the parameters
  $\Longrightarrow$ **Non-parametric optimization**

Programmation génétique

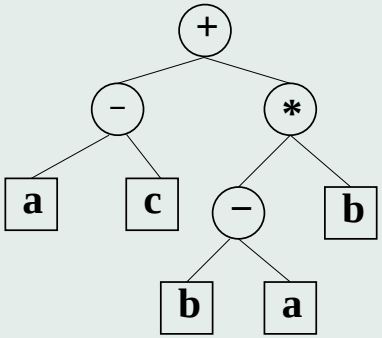
J. Koza – 1992

$\mathcal{F} : \Omega \mapsto \mathbb{R}$ Trouver argmax ($\mathcal{F}$)

Le rêve : Le programme qui écrit le programme

Ω = espace de programmes
$\mathcal{F}$ = qualité d'un programme
S-expressions :

$\mathcal{T}$ = { Vars, Cstes }
$\mathcal{N}$ = { opérateurs }



Quelques applications

- Classification
- Régression symbolique
- Prédiction séries chaotiques

- Stratégies multi-agents (e.g. jeux, ...)
- Robotique
- Génération de plans

- Conception de circuits analogiques
- Apprentissage de réseaux neuronaux
- Modélisation mécanique

Concepts

GP = rejeton de GAs

Traits distinctifs : matériel génétique

structuré (souvent sous forme d'arbre)

de taille variable (bornée)

souvent exécutable

Historique :

Représentation: Langage LISP

Publications: Cramer 85, Koza 89, Koza 92, Koza 94

Remarque : pas d'alternative en optimisation classique

Espaces d'arbres

Etant donné :

Un ensemble $\mathcal{N}$ de noeuds (ou opérateurs)

Un ensemble $\mathcal{T}$ de feuilles (ou opérandes)

Ω = Arbres($\mathcal{N}, \mathcal{T}$)

Exemples :

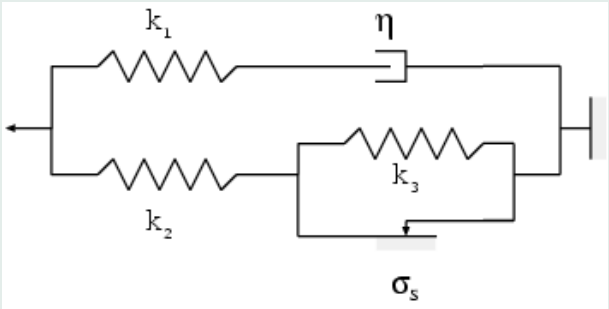
- $\begin{cases} \mathcal{N} = \{+, \times\} \\ \mathcal{T} = \{X, \mathcal{R}\} \\ \Omega = \text{ Polynomes de } X. \end{cases}$

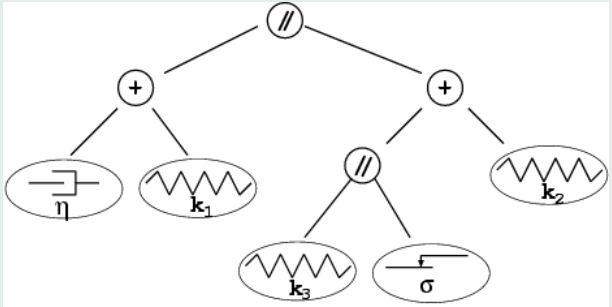
- $\begin{cases} \mathcal{N} = \{ \text{ if-then-else, while-do, repeat-until,..}\} \\ \mathcal{T} = \{\text{expressions, instructions}\} \\ \Omega = \text{ Programmes} \end{cases}$

Rheological GP

Rheological models $\equiv$ Trees built from

- $\mathcal{N} = \{$ series $+$, parallel // $\}$
- $\mathcal{T} = \{$ Spring(k), Slider(σ_S), Dashpot(η)$\}$





Initialisation

Creer Arbre:

Choisir Noeud dans { +, //, ressort, patin, amort }

Si Noeud dans { ressort, patin, amortisseur }

Tirer Constante k, σ_S, η dans $\mathbb{R}^+$

Retour (Noeud, constante)

Si Noeud dans { +, // }

Fils_1 = Creer Arbre

Fils_2 = Creer Arbre

Retour (Noeud, Fils_1, Fils_2)

Evaluation

Compilation

$$H \quad \rightarrow \quad \textit{Système d'équations } \mathcal{S}_H$$

- Ressort(k) $\sigma(t) = k \,.\, \varepsilon(t)$
- Amortisseur(η) $\sigma(t) = \eta \,.\, \dot{\varepsilon}(t)$
- Patin(σ_S) $(\dot{\varepsilon}(t) = 0)\ OR\ (|\sigma(t)| = \sigma_S)$
- Série
  $\varepsilon_{parent}(t) = \varepsilon_{fils_1}(t) + \varepsilon_{fils_2}(t)$
  $\sigma_{parent}(t) = \sigma_{fils_1}(t) = \sigma_{fils_2}(t)$
- Parallèle
  $\varepsilon_{parent}(t) = \varepsilon_{fils_1}(t) = \varepsilon_{fils_2}(t)$
  $\sigma_{parent}(t) = \sigma_{fils_1}(t) + \sigma_{fils_2}(t)$

Simulation

$$\mathcal{S}_H \cup \ (\varepsilon_H(t) = \varepsilon_{exp}(t)) \ \rightarrow \ \sigma_H(t)$$

Evaluation

$$f(H) = Distance(\sigma_H, \sigma_{exp})$$

Critère d'arrêt

Sources d'erreur

- ED $\longrightarrow$ Différences finies
- Erreurs expérimentales
- Bruit de résolution

Estimation de l'erreur

$$Err = ||\sigma_H(t_{exp} = t_1, t_2, t_3, ...) - \sigma_H(t_{exp} = t_1, t_3, t_5, ...)||$$

Critère de succès

$$f(H) \approx Err$$

Conclusion partielle

- **Identification de modèle rhéologique par GP**
  Premiers résultats positifs.
  Passage à l'échelle difficile.

- **C'est un cas favorable :**
  Tout élément de l'espace de recherche est acceptable...

GP and Background Knowledge

- **EC: same evolution as AI**
  - I. A universal tool 1965
  - II. Knowledge makes the difference 1991

- **GP: the closure assumption**
  - any subtree is a valid operand for any operator.

Pros	simple crossover
	simple mutation
Cons	Search space size

GP and Background Knowledge, 2

- **Syntactic constraints**

  Gruau 96, Keijzer Babovic 99

- **Strongly typed GP**

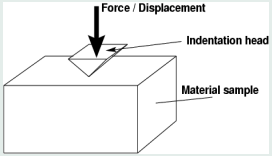
  Montana 97

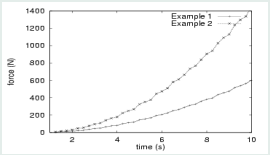
- **BNF Grammars**

  Horner 96, Ryan et al. 98

Dimension aware GP
The mechanical problem

Indentation experiments on unknown material





Goal: Find expression $\mathcal{F}$ s.t.

$$\text{Force} = \mathcal{F}(\text{displacement, time, material parameters})$$

Ratle, Sebag, 2000

GP and Machine Discovery

Trivial BK: Dimension-consistency

meters + seconds ? Oups !

Assumption:

finite set of units	$\{m, s, kg\}$
compound units	$U_{ijk} : m^i s^j kg^k$
limited combinations	$i, j, k \in [-2, 2]$

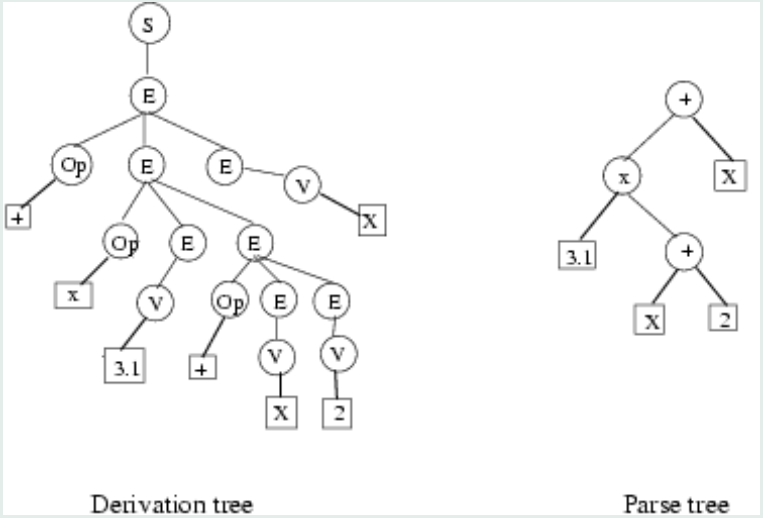
Representation: BNF grammars

S	start symbol	$U_{1,-2,1}$
N	non-terminals	$\{U_{ijk}\}$
T	terminals	$\{Vars, \mathcal{R}, +, -, *, /, exp\}$
P	production rules	

$$
\begin{aligned}
U_{ijk} := \quad & U_{ijk} + U_{ijk} \mid U_{ijk} - U_{ijk} \mid U_{ijk}\ exp^{\,U_{000}} \\
& \mid_{abc+def=ijk} U_{abc}\ *\ U_{def} \\
& \mid_{abc-def=ijk} U_{abc}\ /\ U_{def} \\
& \mid_{unit(var)=ijk} Var
\end{aligned}
$$

Enforcing constraints through grammars

Derivation trees $\longrightarrow$ Parse trees



Beware !	**CFG**	**GP**
Terminals	variables, constants, operators	variables and constants
Non-Terminals	typed expressions	operators

GP on derivation trees – Gruau 96

- Initialization: uniform selection among derivations in a production rule

  filter out trees with depth $> D_{max}$

- Crossover: swap nodes with same non-terminal symbol

  $\equiv$ Strongly Type Genetic Programming

  Montana 1995, Haynes et al. 1996

- Mutation: select another derivation

Dimension grammar

Physical units			
Quantity	mass	length	time
Variables			
K (Elastic element)	+1	0	–1
n (Viscous element)	+1	0	–1
t (time)	0	0	+1
u (displacement)	0	1	0
Solution			
F (Force)	1	1	-2

Automatic generation of the grammar

each compound unit → a non-terminal symbol

admissible combinations → production rules

N	non-terminals	$\{U_{ijk}\}$
T	terminals	$\{Vars, \mathcal{R}, +, -, *, /, exp\}$
P	production rules	

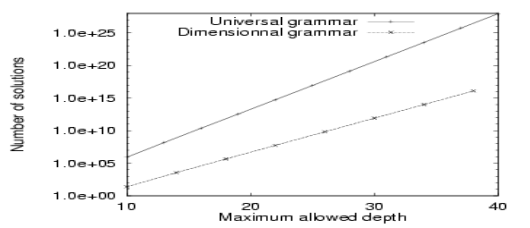
$$
\begin{aligned}
U_{ijk} := \; & U_{ijk} + U_{ijk} \mid U_{ijk} - U_{ijk} \mid U_{ijk}\, exp^{\,U_{000}} \\
& \Big|_{abc+def=ijk} U_{abc} \; * \; U_{def} \\
& \Big|_{abc-def=ijk} U_{abc} \; / \; U_{def} \\
& \Big|_{unit(var)=ijk} Var
\end{aligned}
$$

$$
\mathcal{F} := \quad mass \times length \times time^{-2}
$$

Automatically generated

First Results

Reduction of the search space



Poor performances ...blamed on Initialization

Uniform initialization: $\mathcal{P}$(non-terminal) $\gg \mathcal{P}$ (terminal)

deep trees, most are filtered out

Note : Similar to constrained optimization with sparse feasible region

Ryan et al, 1998

Poor initial population $\rightarrow$ poor performances

Initialization in Grammar Guided GP

Biased initialization fails

- Set $\mathcal{P}$(terminals) $\gg$ $\mathcal{P}$(non-terminals)
- Population poorly diversified, premature convergence

Constraint resolution for initialization

- Minimal tree depth for each non-terminal or derivation
- On-line filtering out of derivations

  incompatible with maximum depth

- GP initialization = constraint solver

$\longrightarrow$ Diversified initial population within depth D_{Max}

Constrained Initialization for Grammar-Guided GP

- **Compute $d_{min}(U)$ = U minimal depth**

$$U := deriv_1 \mid \ldots \mid deriv_N$$
$$d_{min}(U) = min_i\, d_{min}(deriv_i)$$
$$d_{min}(U_1\ op\ U_2) = 1 + max(d_{min}(U_1), d_{min}(U_2))$$

Constrained Initialization, 2

- **Construct Exp with maximal depth D_{Max}**

  $Exp = S$; $\quad d_{max}(S) = D_{Max}$

  While (exists non terminal symbols in Exp)

  Select U in Exp $\quad U = |_i\ deriv_i$

  Select $deriv_i$ / $d_{min}(deriv_i) \leq d_{max}(U)$

  $deriv_i = U_1\ op\ U_2$

  Set $d_{max}(U_1) = d_{max}(U_2) = d_{max}(U) - 1$

- **Result:**

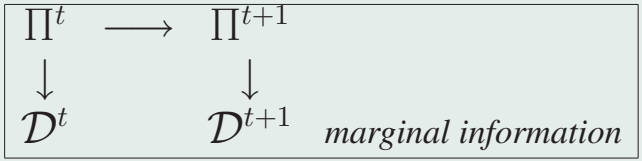
  admissible and diversified individuals

Can we learn more ?

Populations $\Pi \longleftrightarrow$ **Distributions** $\mathcal{D}$

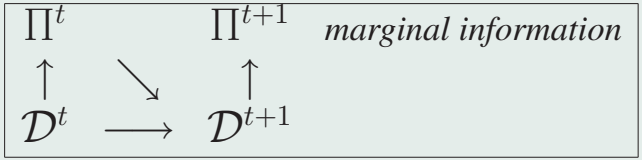
Evolutionary Computation

works in extension



An AI approach: PBIL Baluja 1995

works in intension



Probabilistic Grammar-Guided GP

Principle: Setting weights on each derivation

Scal: $deriv_i \rightarrow w(deriv_i)$

Vect: $deriv_i \times depth\ k \rightarrow w(deriv_i,\ k)$

Salustowicz & Schmidhuber, 1998

Initializing Distribution

$\forall\, i, \forall\, k,\ \ w(deriv_i,\ k) = 1$

Generating Individuals

for U at depth k, if $d_{min}(deriv_i) \leq d_{max}(U)$

Prob(Select $deriv_i$) $\propto w(deriv_i,\ k)$

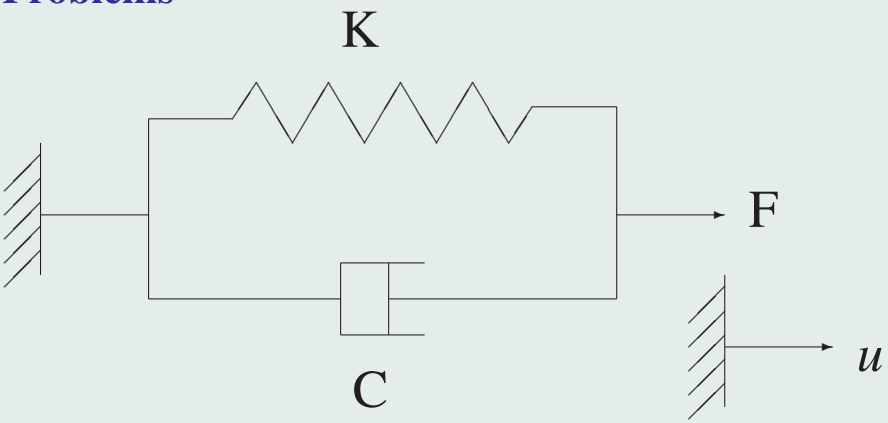
Probabilistic Grammar-Guided GP, 2

Updating Distribution

Loop on the best individuals
  if $deriv_i$ is chosen at depth k
    $w(deriv_i,\ k) *= (1 + \epsilon)$
Loop on the worst individuals
  if $deriv_i$ is chosen at depth k
    $w(deriv_i,\ k) *= (1 - \epsilon)$

Empirical Validation

Problems



$$x(t) = \frac{F}{K}\left(1 - e^{\frac{-Kt}{C}}\right) \tag{1}$$

$$x(t) = \frac{F}{K_1} + \frac{Ft}{C_1} + \frac{F}{K_2}\left(1 - e^{\frac{-Kt}{C_2}}\right) \tag{2}$$

Empirical Validation, 2

Grammars

Universal $S := NT$

$NT := T \mid OP\ NT\ NT$

$OP := + \mid - \mid * \mid \div \mid exp$

$T := F \mid K \mid C \mid t \mid 1 \mid 2 \mid 3 \mid 4$

Universal + exp-neg: same as above, except

$OP := + \mid - \mid * \mid \div \mid exp \mid exp-$

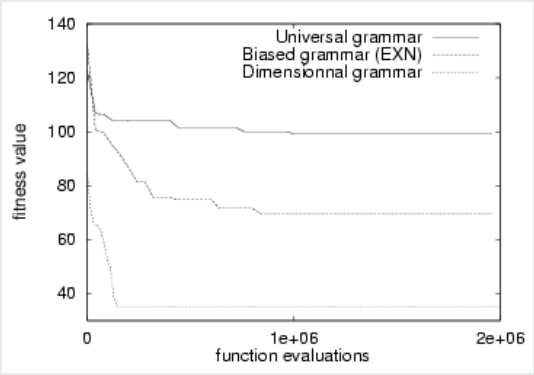
Dimensional: dimension-consistent grammar

Physical units			
Quantity	mass	length	time
Variables			
E (Force)	+1	+1	–2
K (Elastic element)	+1	0	–1
n (Viscous element)	+1	0	–1
t (time)	0	0	+1
Solution			
x (displacement)	0	+1	0

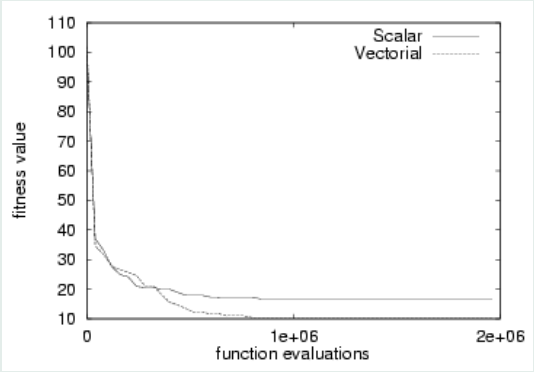
Empirical Validation, Parameters

Parameter	Value	
Algorithm	GP	GPwPG
Population size	2000	500
Max. number of generations	1000	4000
Probability of Crossover	0.8	—
Probability of tree mutation	0.2	—
Probability of point mutation	0.8	—
Nb of best individuals for learning	—	2
Nb of worst individuals for learning	—	2
Learning rate (ϵ)	—	0.001
Probability of perturbation	—	0.001
Amplitude of perturbation	—	0.001
Number of training examples	20	20
Number of independent runs	10	20

Results



Convergence: Impact of grammars



Convergence: Impact of learning distributions

Philosophie générale

Pour l'homme qui a un marteau, tout ressemble à un clou...

Ne pas avoir d'algorithme favori

Le meilleur algorithme dépend du problème

Entre science et technologie

ne pas faire l'autruche mais

réaliser la difficulté des résultats négatifs

Regarder au dehors

Vos voisins de labo ont souvent des problèmes voisins

et des façons différentes de les regarder...

Valider

Théoriquement *et* pratiquement