21 mars 2005

Apprentissage statistique et Optimisation stochastique

Michèle Sebag

TAO

CNRS – INRIA – LRI

http://tao.lri.fr

Plan

- Apprentissage
- Apprentissage : programmation par les données
- Optimisation
- Optimisation : programmation par les préférences

Apprentissage supervisé

Input

- Données brutes : signaux, données expérimentales, logs d'EGEE, ...
- Etiquettes associées : signaux positifs/négatifs,
  jobs DONE, LOST, ABORT

Base d'apprentissage

$$\mathcal{E} = \{(x_i, y_i), x_i \in X, y_i \in Y\}$$

$Y = \mathbb{R}$ régression
$Y = \{1, -1\}$ classification

Apprentissage supervisé, 2

$$\mathcal{E} = \{(x_i, y_i), x_i \in X, y_i \in Y\}$$

Output : Hypothèses

$$h : X \mapsto Y$$

Critères

- Qualité prédictive

ℓ = fonction de perte

$$h^* = Argmin\{E[\ell(h(x), y)],\ h\ \in\ H\}$$

- Compréhensibilité

h est intelligible, l'expert comprend et agit

Programmation par Apprentissage

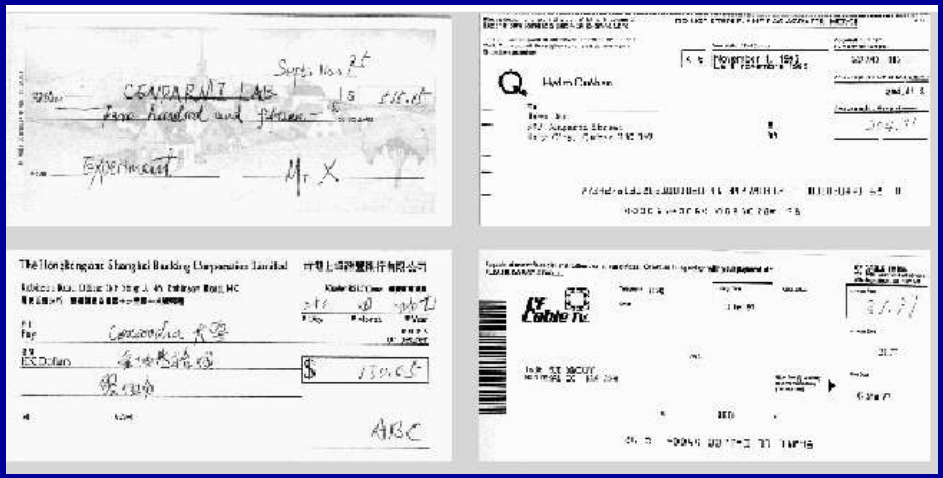
Trois exemples

Dietterich, Oregon State University

- Hand-written recognition
- Wafer test
- Camera allocation in autonomous robotics

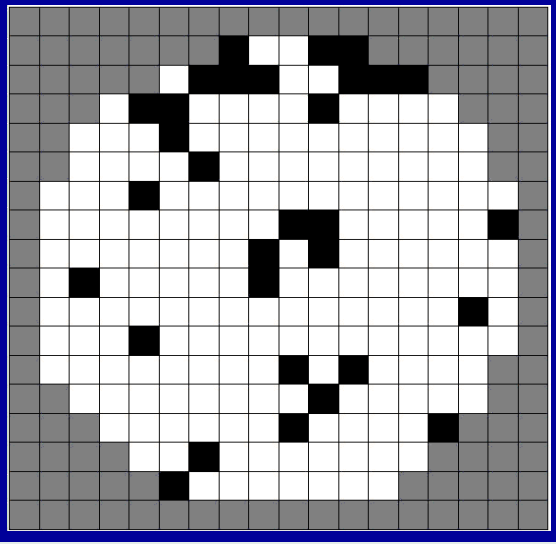
Scenario 1: Reading Checks

Find and read “ courtesy amount” on checks:



Scenario 2: VLSI Wafer Testing

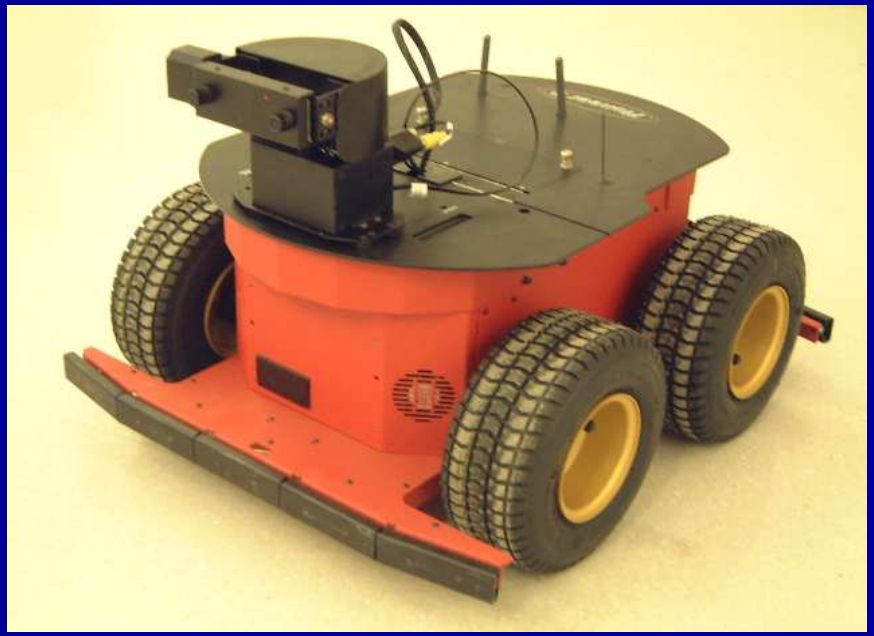
- Wafer test: Functional test of each die (chip) while on the wafer



Scenario 3: Allocating mobile robot camera

Binocular

No GPS



Programmation classique

Analyse : Interviewer experts et utilisateurs pour identifier quoi et comment faire

Conception : Programmer

Implementer

Tester

Limites : Spécifications ardues; compromis mal définis

Spécifications ardues

Absence d’expertise
fraude téléphonique

Expertise implicite
reconnaissance de caractères

Spécifications évolutives
détection d’intrusion

Personnalisation
filtrage de pourriels

Compromis mal définis

Cas du waffer testing

dépend de la probabilité de panne, du coût du test (Wafer vs SIP)

Cas de la caméra robotique

dépend de la probabilité des obstacles, et de la qualité des amers.

Programmation par Apprentissage

Remplacer les choix du programmeur

reconnaissance de caractères, point d'équilibre pour le test, ...

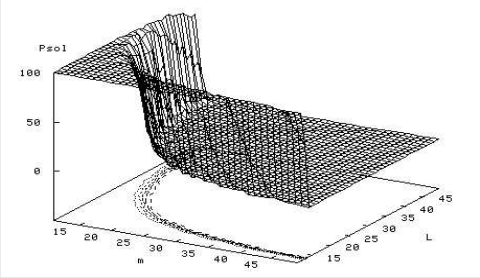
...par l'apprentissage / la fouille de données

- utilisant des modèles appris à partir des données

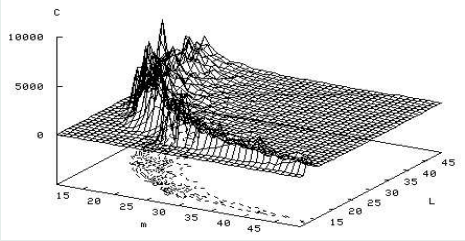
Défi: utiliser la connaissance du domaine

Transitions de phase

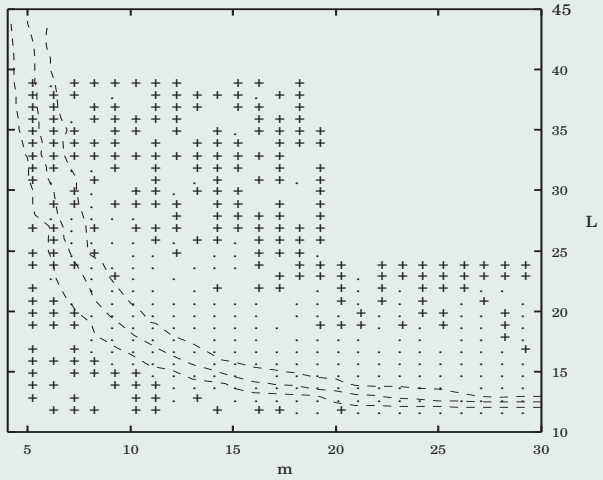
Probabilité de couverture(m, L)



Complexité effective(m, L)



Learning Competence Map



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

$n = 4, N = 100$

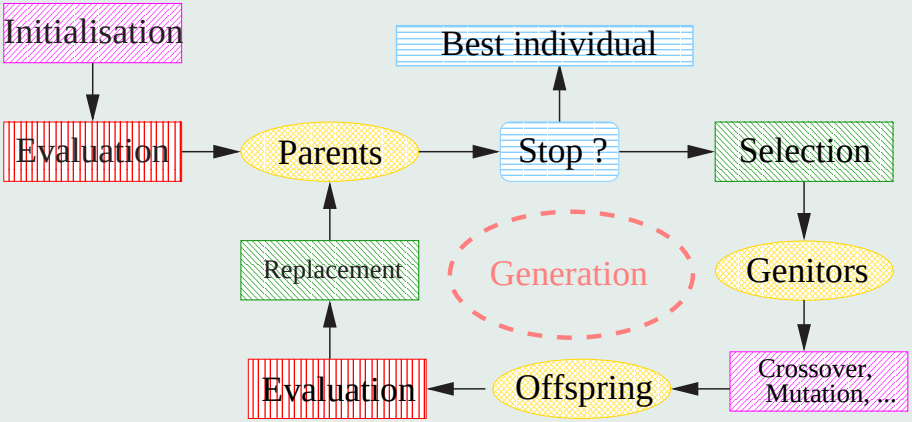
Plan

- Apprentissage
- Apprentissage : programmation par les données
- Optimisation
- Optimisation : programmation par les préférences

Stochastic population-based Optimisation

Evolutionary Computation

$$\mathcal{F} : \Omega \mapsto \mathbb{R}, \quad \text{Find ArgMax}(\mathcal{F})$$



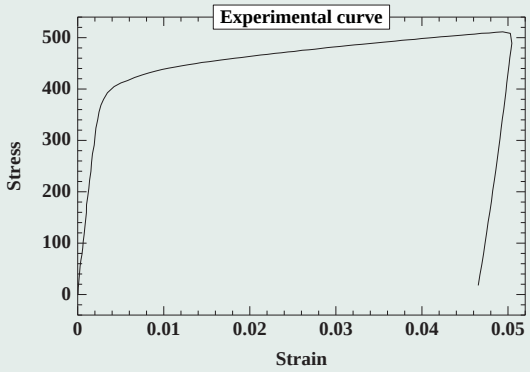
Evolutionary Computation: When ?

$$\mathcal{F} : \Omega \mapsto \mathbb{R}, \quad \text{Find ArgMax}(\mathcal{F})$$

Aimed at ill-posed problems:

- Dynamic $\mathcal{F}$
- Chaotic $\mathcal{F}$
- Mixed search spaces
- Infinite search spaces
- Non computable $\mathcal{F}$

Exploring the expert's search space
Identification of Behavioral Laws



Input: Experimental curves

Identification of Behavioral Laws, 2

Input: Experimental curves

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

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 $\sigma(t) = c.\epsilon(t) + d.\dot{\epsilon}(t)$

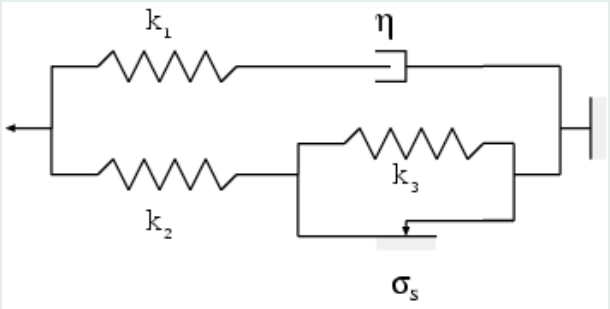
Criteria: the law must fit the experiments **and** be comprehensible.

Search space: Rheological models

Dynamic 1-D laws.

Assembly in series or parallel of

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



Genetic Programming

- Stochastic optimisation

  initialization, (selection, crossover, mutation)$*$

- Search space = Trees ($\mathcal{N}$, $\mathcal{T}$)

  Polynoms $\quad \mathcal{N} = \{+, -, *\}$
  $\mathcal{T} = \{X_1, X_2, \ldots, X_n, \mathcal{R}\}$

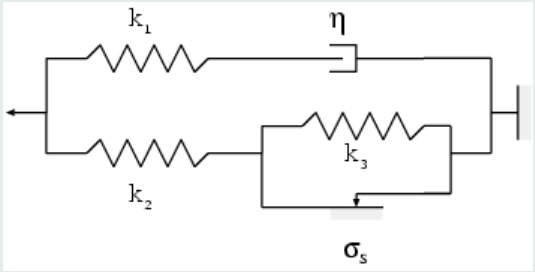
  Programs $\quad \mathcal{N} = \{\text{if-then-else, do-while}\}$
  $\mathcal{T} = \{\text{expressions, instructions}\}$

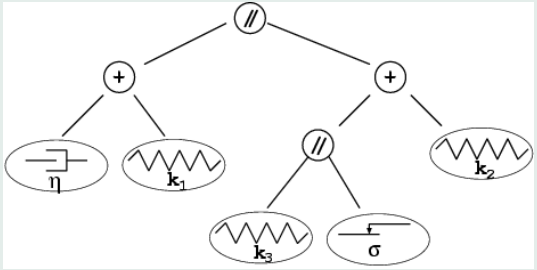
- Fitness : fitting experimental data

Non parametric identification with GP

- Rheological models = Trees $(\mathcal{N}, \mathcal{T})$

$\mathcal{N} = \{//, +\}$ $\mathcal{T} = \{\text{springs, sliders, dashpots }\}$

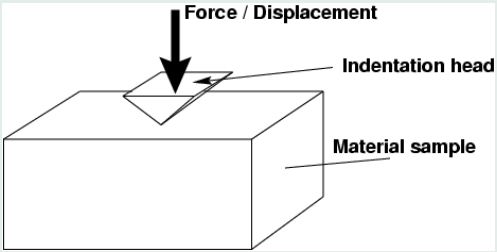


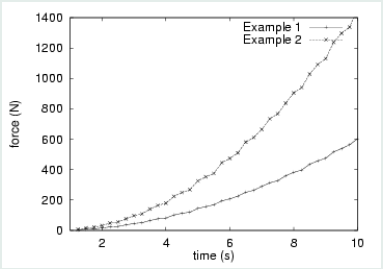


Non parametric Identification

with Background Knowledge

- Macro-Mechanical Identification, Indentation laws





GP and Background Knowledge
ex: Dimension consistency

meters and seconds should not be added...

Goal: Enforcing dimension consistency

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	

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

Plan

- Apprentissage
- Apprentissage : programmation par les données
- Optimisation
- Optimisation : programmation par les préférences

Understanding the User's Goal

Functional Brain Imagery

- Patients, Experiments, Measures

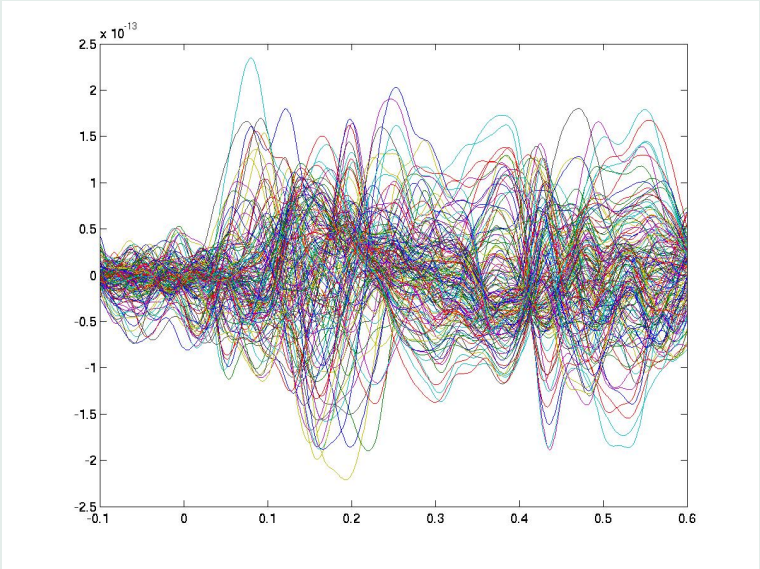
EEG

MEG

- Magneto-Encephalography

1,000 measures *per* sensor *per* second.

The data



Spatio-temporal structure

- Sensors $i = 1..N$
- $i \quad \rightarrow \quad \begin{cases} M_i = (x_i, y_i, z_i) \ \in \mathbb{R}^3 \\ \{C_i[t], t = 1..T\} \ \in \mathbb{R}^T \end{cases}$

Goal

Find spatio-temporal patterns

- Spatial region $A \subset \mathbb{R}^3$
- Temporal interval $I \subset \{1..T\}$

  defining

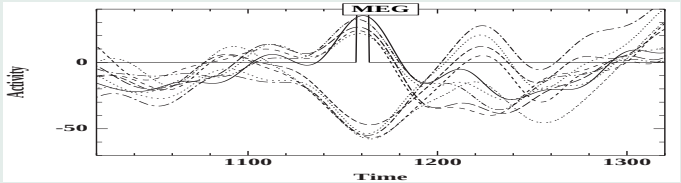
$$\mathcal{V}(A, I) = \{C_k[t],\ k\ \in\ A,\ t\ \in I\}$$

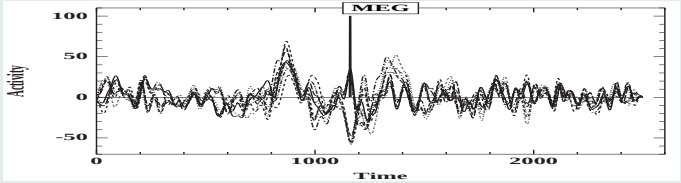
SUCH THAT

the variance of signals within $\mathcal{V}(A, I)$ is low
and $A \times I$ is a large spatio-temporal region

"active areas of the brain"

Discriminant Spatio-Temporal Patterns





What’s difficult ?

Solve the problem

Understand the problem

Understand the expert’s goals

From Xtrem Programming to Xtrem Solving