VCD bounds for some GP genotypes

José Luis Montaña
Departament of Mathematics, Statistics and Computer Sciences
University of Cantabria (Spain)

Vapnik-Chervonenkis Dimension

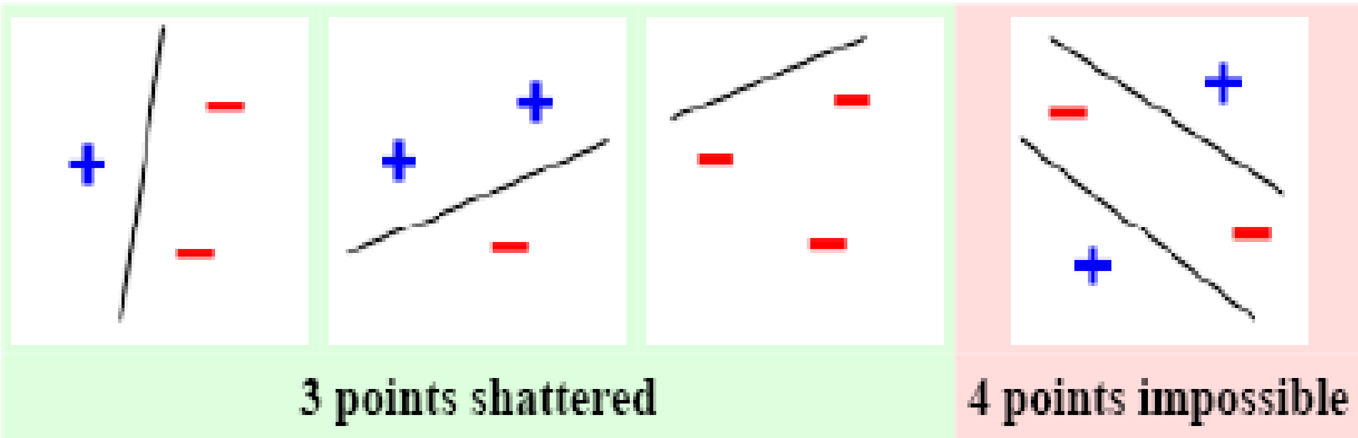
- In computational learning theory, the **VC dimension** (for **Vapnik - Chervonenkis dimension**) is a measure of the capacity of a statistical classification algorithm, defined as the cardinality of the largest set of points that the algorithm can shatter. It is a core concept in Vapnik-Chervonenkis theory, and was originally defined by Vladimir Vapnik and Alexey Chervonenkis.

- V. Vapnik and A. Chervonenkis. "On the uniform convergence of relative frequencies of events to their probabilities." *Theory of Probability and its Applications*, 16(2):264--280, 1971.
- A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. "Learnability and the Vapnik-Chervonenkis dimension." *Journal of the ACM*, 36(4):929--865, 1989.

Shattering

- A classification model $f(x,\alpha)$ with some parameter vector a is said to *shatter* a set of data points $(x_1,\ldots,x_m)$ if, for all assignments of labels to those points, there exists an α such that the model *f* makes no errors when evaluating that set of data points.
- **VC dimension** of a model *f* is *h* where *h* is the maximum *h* such that some data point set of cardinality *h* can be shattered by *f*.

Shattering (cont.)



Interpretation

- The VC dimension has utility in learning theory, because it can predict a probabilistic upper bound on the test error of a classification model.
- The bound on the test error of a classification model (on data that is drawn i.i.d. from the same distribution as the training set) is given by

$$\varepsilon(\alpha) \leq \varepsilon_m(\alpha) + \sqrt{\frac{h(log(2m/h)+1) - log(\eta/4)}{m}},$$

- with probability 1 − η, where *h* is the VC dimension of the classification model, and *m* is the size of the training set (restriction: this formula is valid when the m dimension is large enough, *h* < *m*).

VC dimension vs. Syntactical representation

Syntactical representation	Invariants	VC Dimension
Polynomials	Degree **d**, number of variables **n**	$\mathbf{O(n^d)}$
First order formulas over the reals	Formula size **s**, degree of the polynomials **d**, number of constants **k**	**2k log (4eds)** [Goldberg&Jerrum, 95]
Neuronal networks	Number of programable parameters **k**	$\mathbf{O(k^2)}$ [Karpinski&Macintyre 97]
GP trees (representing computer programs, more generaly symbolic expressions)	Computational complexity of the program, number of variables, …	Length, space complexity, size, etc.

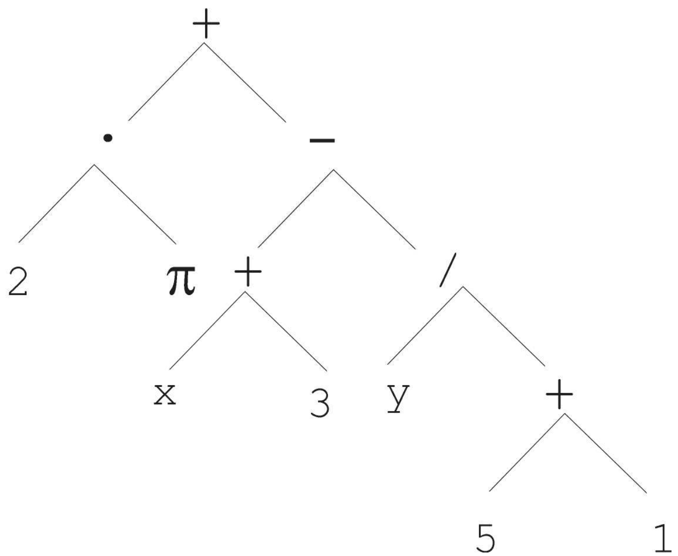
Symbolic expressions

- Symbolic expressions can be defined from
  - Terminal set T
  - Function set F (with the arities of function symbols)
- Adopting the following general recursive definition:
  - Every $t \in T$ is a correct expression
  - $f(e_1, \ldots, e_n)$ is a correct expression if $f \in F$
  - arity(f)=n and $e_1, \ldots, e_n$ are correct expressions

  There are no other forms of correct expressions

GP-trees: Tree based representation of symbolic expressions

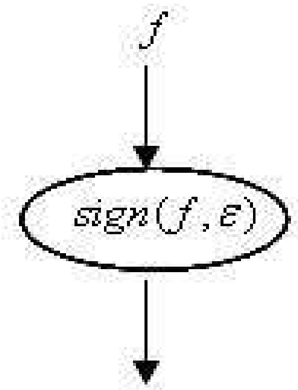
- Rational functions:
  - Terminals: variables and the real constants.
  - Functionals : arithmetic operations {+,-,x,/}.



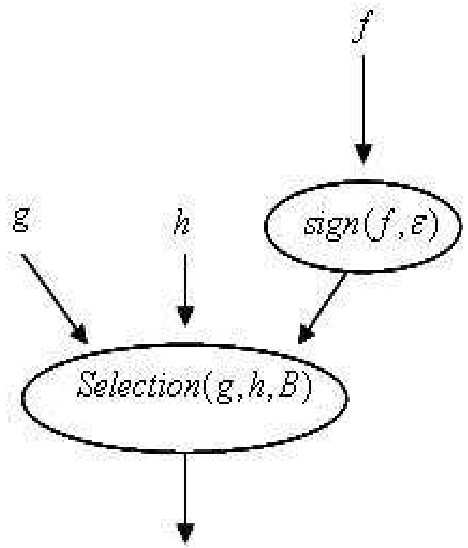
$$2\cdot\pi+\left((x+3)-\frac{y}{5+1}\right)$$

Tree based representation of symbolic expressions (cont.)

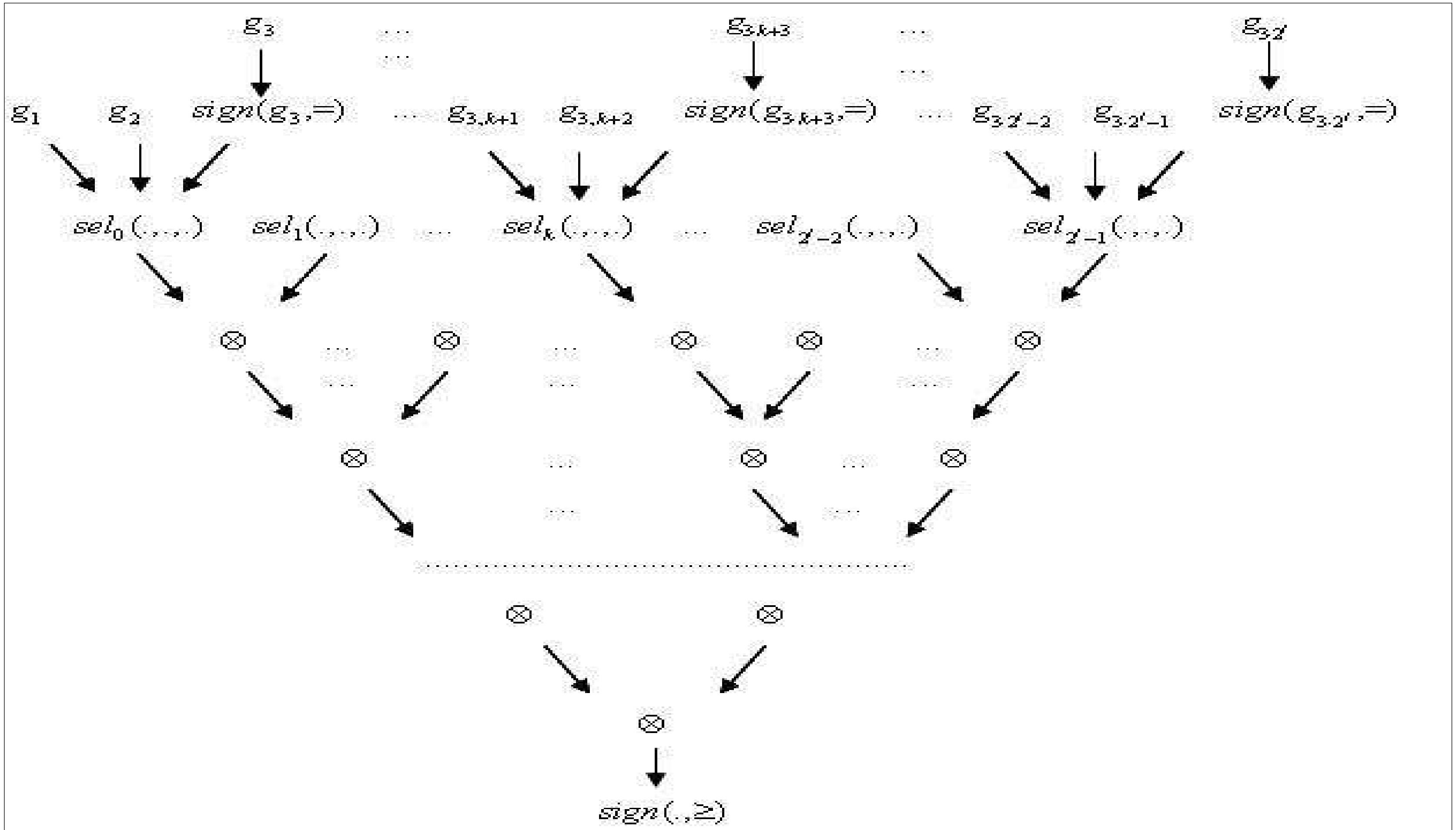
- Straight line programs.
  - Terminals: variables and real constants.
  - Functionals: arithmetic operations, root extraction,…, sign tests, if (-) then {-} else{-} instructions,



$\varepsilon \in \{\geq, \leq =, >, <\}$



Tree representation of straight line programs: Tree T(l).



Tree representation of straight line programs (cont.)

Tree size	Tree height	VC dimension
Sequential Complexity	Parallel complexity	O(log D+ log S) D=degree S=formula size
Tree T(l) $\theta(2^l)$	Tree T(l) $\theta(l)$	Tree t(l) Best bound $O(2^l)$

Tree representation of straight line programs (cont.)

- C(l) concept class defined by T(l).
- Formula describing C(l):

$$\begin{aligned}\Phi(l) = (&g_3 = 0, g_6 = 0, \ldots, g_{3.2^l} = 0, g_1 g_4 \cdots g_{3.2^l - 2} \geq 0) \vee \cdots \\ \cdots \vee (&g_3 \neq 0, g_6 \neq 0, \ldots, g_{3.2^l} \neq 0, g_2 g_5 \cdots g_{3.2^l - 1} \geq 0)\end{aligned}$$

Best upper bound for VCD of C(l) is $O(2^l)$.

Tree representation of straight line programs (cont.)

- ***Lemma*** (Based on [Grigoriev 88], [Fitchas et al. 87]).
- F: family of n variate polynomials with real coefficients.

$$D = \sum_{f \in F} \deg(f)$$

- Then, the number of consistent sign assigments (f>0,f=0,f<0) to polynomials of the family F is at most:

$$(1 + D)^n$$

Tree representation of straight line programs (cont.)

Corollary. **VCD of C(l) is O(l).**

Proof. Use previous lemma and the fact:

$$\Phi(l) = \vee_i (\Phi_{i,1}(l) \wedge \Phi_{i,2}(l))$$

$$\Phi_{i,1}(l): \quad sign\ \ assig \quad g_3, \ldots, g_{3.2^l}$$

$$\Phi_{i,2}(l): \quad \Pi_j\, g_{k(j)} \geq 0$$

Formula size of GP-trees representing straight line programs

- ***Lemma (main)***.
- $C_{k,n}$: family of concept classes whose memebership test can be represented by a family of trees
- $T_{k,n}$ with k constants and n variables $C_{k,n.}$
- h=h(k,n) height of $T_{k,n}$.

Then

- $C_{k,n}$ has formula size $2^{(k+n)h^2}$
- Degree of polymomials 2^{h}
- ***Interpretation***. **Formula size does not depends on sequential time but on parallel time (i. e height of the GP-tree).**

VCD of GP-trees representing straight line programs

- ***Theorem (main)***.
- $C_{k,n}$: family of concept classes.
- $T_{k,n}$: family of trees. (membership test to $C_{k,n}$)
- h=h(k,n) depth of $N_{k,n}$.

Then

- $C_{k,n}$ has VC dimension $\boxed{O(k(k+n)h^2)}$
- ***Interpretation***. **VC dimension depends polynomially on parallel time.**

VCD regularization for model selection in GP

- Symbolic regression under the general setting of predictive learning (Vapnik 95, Cherkassy & Mulier 98,…).
- Estimate unknown real-valued function

$$y=g(x)$$

- x is a multidimensional input and y is an scalar output.

VCD regularization for model selection in GP (cont.)

- The estimation is made based on a finite number of samples (training data) (x_i,y_i) $(i=1,\ldots,m)$ i.i.d generated according to someunknown joint probability distribution:

$$p(x,y)=p(x)\ p(y|x)$$

- According to SLT the unknown function (regression function) is
- Mean value of the output conditional probability:

$$g(x)=\int y\ p(y|x)\ dy$$

VCD regularization for model selection in GP (cont.)

- A learning method selects the best model (concept) $f(x,\alpha_0)$ from a set of possible models (concept class)

$$\{f(x,\alpha): \alpha \in \Theta\}$$

- The quality of a model $f(x,\alpha)$ is measured by the mean square error.

$$\varepsilon(\alpha) = \int (y - f(x,\alpha))^2 \, p(x,y) \, dx \, dy \quad [RF]$$

- Learning is the problem of finding the model $f(x,\alpha)$ that minimizes the risk functional [RF].

Empirical Risk Minimization

- For a given parametric model with finite VC dimension the model parameters are estimated by minimizing the empirical risk:

$$\varepsilon_m(\alpha) = 1/m \sum_{i=1,\ldots,m} (y_i - f(x_i, \alpha))^2$$

- ERM is founded in the formula:

$$\varepsilon(\alpha) \leq \varepsilon_m(\alpha) + \sqrt{\frac{h(log(2m/h)+1) - log(\eta/4)}{m}},$$

- Examples: select a degree d polynomial, select a linear regressor with fixed number of parameters, select a computer program of bounded complexity, etc.

Structural Risk Minimization

- The problem of model selection appears when VCD of the set of possible models is infinite.
- Examples: select a polynomial, a formula, a linear regressor with unbounded number of parameters, a computer program, a GP tree,…All these genotypes have infinite VCD.

Structural Risk Minimization with VC dimension

- Under SRM a set of possible models V forms a nested structure
- $V_1 \subseteq V_2 \subseteq V_3 \subseteq \ldots. \subseteq V_h \subseteq \ldots$
- Each element V_h represents the set of models of complexity bounded by h.
- VC dimension is an increasing function on h.
- Select the model minimizing:

$$\varepsilon_m(\alpha).\left(1-\sqrt{p(\alpha)-p(\alpha)\ln p(\alpha)+\frac{\ln m}{2m}}\right)^{-1}$$

where $p(\alpha)=1/h(\alpha)$ and $h(\alpha)=\min\{h: f(x,\alpha)\in V_h\}$

Structural Risk Minimization with VC dimension for GP

- V= set of all straight line programs with fixed number of terminals (n variables, k constants).
- V_h= set of all straight line programs that can be represented by GP-trees of heights bounded by h.
- Fittness function for a tree T:

$$fitness(T) = empirical\ risk\,(T).\left(1 - \sqrt{p(T) - p(T)\ln p(T) + \frac{\ln m}{2m}}\right)^{-1}$$