GRAPH MACHINES AND THEIR APPLICATIONS TO COMPUTER-AIDED DRUG DESIGN: A NEW APPROACH TO LEARNING FROM STRUCTURED DATA

A. GOULON-SIGWALT, A. DUPRAT, G. DREYFUS
ESPCI, Laboratoire d'Électronique
10 rue Vauquelin - 75005 PARIS

Phone: (33) (0)1 40 79 45 41 - (33) (0)6 14 09 57 74
Gerard.Dreyfus@espci.fr - http://www.neurones.espci.fr

OUTLINE

CONVENTIONAL MACHINE LEARNING: AN OVERVIEW

LEARNING NUMBERS FROM GRAPHS: GRAPH MACHINES

MODEL SELECTION FOR GRAPH MACHINES: VIRTUAL LEAVE-ONE-OUT

APPLICATION TO COMPUTER-AIDED DRUG DESIGN

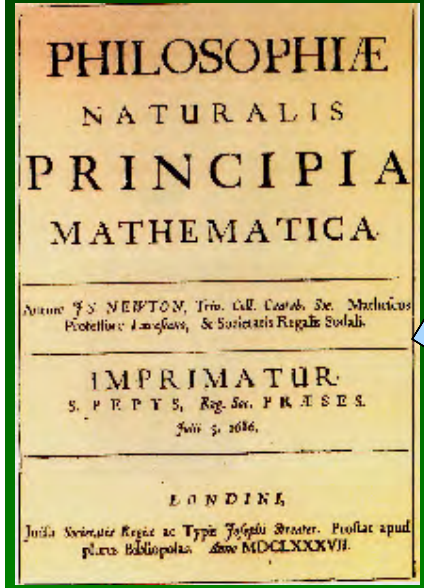
CONVENTIONAL “SUPERVISED” MACHINE LEARNING

PURPOSE: LEARN, *FROM EXAMPLES*, A MAPPING FROM A SPACE OF VARIABLES (“FEATURE SPACE”) TO A “TARGET” SPACE.

EACH *EXAMPLE k* IS A COUPLE:

- A VECTOR OF VARIABLES (OR FEATURES) $\mathbf{x}^k$,
- THE CORRESPONDING “TARGET” VALUE y_p^k (TYPICALLY A SCALAR).

KNOWLEDGE-BASED MODELING

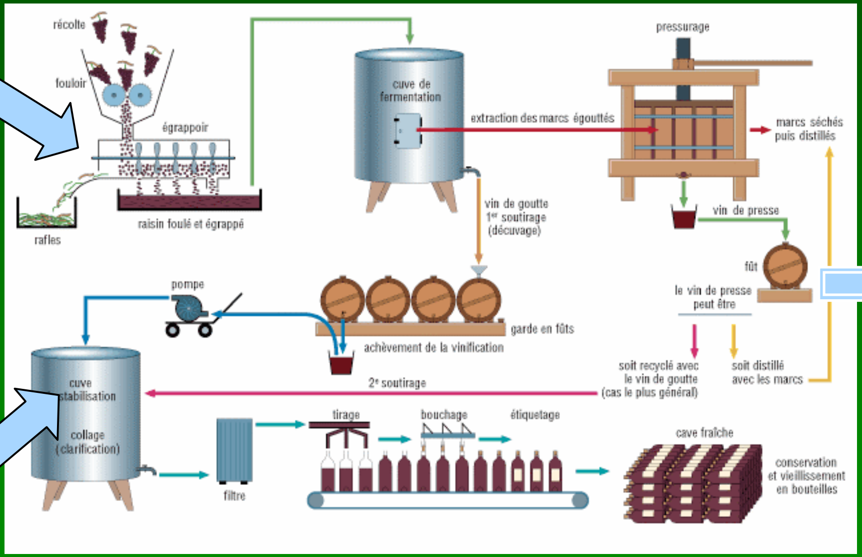


PHILOSOPHIÆ
NATURALIS
PRINCIPIA
MATHEMATICA.

Autore JS. NEWTON, Trin. Coll. Cantab. Soc. Matheſeos
Profeſſore Lucaſiano, & Societatis Regalis Sodali.

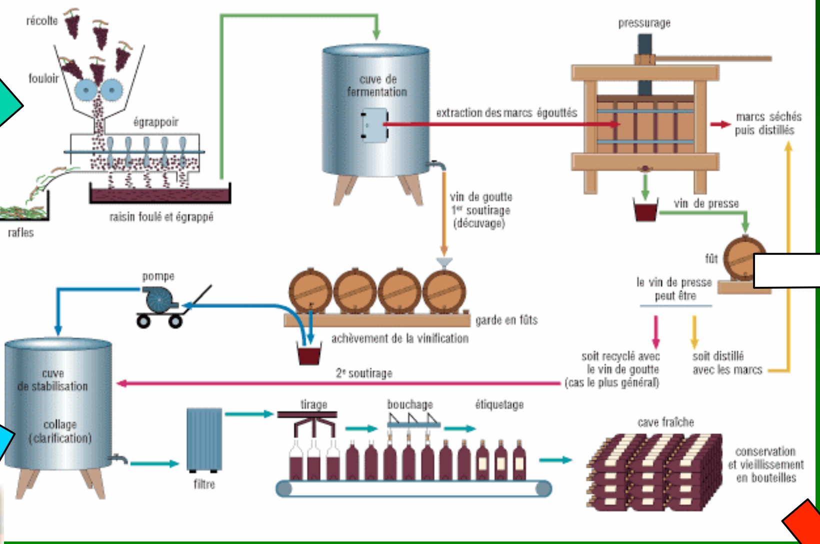
IMPRIMATUR.
S. PEPYS, Reg. Soc. PRÆSES.
Julii 5. 1686.

LONDINI,
Juſſu Societatis Regiæ ac Typis Joſephi Streater. Proſtat apud
plures Bibliopolas. Anno MDCLXXXVII.



CONVENTIONAL MODELING BY MACHINE LEARNING

Nonlinear, parameterized « black-box » model, designed by training from examples



TECHNOLOGIE

Des réseaux de neurones pour juger de la qualité des vendanges

Issus des recherches dans le domaine de l'intelligence artificielle, les réseaux de neurones permettent de multiples applications, y compris pour l'œnologie.

Meilleur est le raisin, meilleur sera le vin. Une assertion qui a tout l'air d'une évidence. Et pourtant… Si le vin est très surveillé tout au long de son élaboration, le produit de base, le raisin, l'est beaucoup moins. Notamment dans les grandes propriétés ou dans les caves regroupant beaucoup de producteurs. « *C'est très paradoxal. La vendange est souvent très mal mesurée, et l'on connaît mal le raisin qui entre dans les chais* », assure Matthieu Dubernet, du laboratoire d'œnologie Dubernet à Narbonne.

La période des vendanges est pourtant une période très sensible. Comme la plupart des fruits, le raisin subit les agressions de la nature, et les décisions doivent être prises rapidement. Traditionnellement, à moins d'une analyse chimique longue et coûteuse, seul l'œil d'un spécialiste est capable de détecter un problème. Aujourd'hui, dans plusieurs dizaines de caves en France et à l'étranger, le processus est automatisé grâce à un système baptisé Grapscan, mis au point par le laboratoire Dubernet. Il repose sur un instrument de spectrophotométrie de la société danoise Foss, travaillant à base d'analyse infrarouge à transformée de Fourier (IRTF). L'appareil, déjà largement utilisé dans l'industrie agroalimentaire, notamment celle du lait, mesure très précisément la quantité de certains composés organiques des moûts (taux de sucre, acidité lactique, acidité tartrique…) qui indiquent la maturité du raisin. Il mesure aussi la présence des micro-organismes parasites (mycéliums, levures et bactéries) qui causent des dégâts redoutés des vignerons : pourriture grise, pourriture acide et fermentation non maîtrisée.

Décomposer les problèmes

Tout le problème, c'est que ces dernières données ne sont pas exploitables en l'état, assure Matthieu Dubernet : « *En regardant tous ces chiffres correspondant au métabolisme des parasites, un expert saura qu'il existe un problème parasitaire, mais il ne connaîtra pas son importance. Le passage entre le constat analytique avec ses chiffres bruts et la réalité biologique est extrêmement complexe.* »

C'est là que les réseaux de neurones entrent en piste, pour offrir une lecture compréhensible malgré la sécheresse des chiffres. Pour comprendre leur intérêt, il faut revenir un peu en arrière. Les recherches sur les réseaux de neurones sont nées avec la neurophysiologie, une science qui a tenté d'expliquer le fonctionnement du système nerveux par modélisa-

Le système Grapscan (en médaillon) utilise les réseaux de neurones pour analyser le raisin récolté.

AFP

peu doué et procède par itérations : plutôt que de trouver une formule magique pour l'ensemble de la courbe, il applique de point en point des formules mathématiques plus simples. C'est le rôle de chacun des neurones. « *On*

neurones en un logiciel capable d'apprentissage lorsqu'on le connecte à une base de données. Ainsi équipé, le réseau de neurones offre une représentation acceptable de la réalité, insiste Patrice Kiener : « *Nous sommes*

comprend pas. Grâce à son historique, le réseau de neurones repère ces interactions et indique un résultat », précise Patrice Kiener.

Le système Grapscan fonctionne ainsi avec trois réseaux de neurones fonctionnant sur un

fonctionne bien. Il n'est pas nécessaire d'aller à l'infini. Ce qui est important, c'est d'améliorer la qualité de ces mesures », insiste Matthieu Dubernet. Les trois passent dans leur Moulinette le même jeu de données correspon-

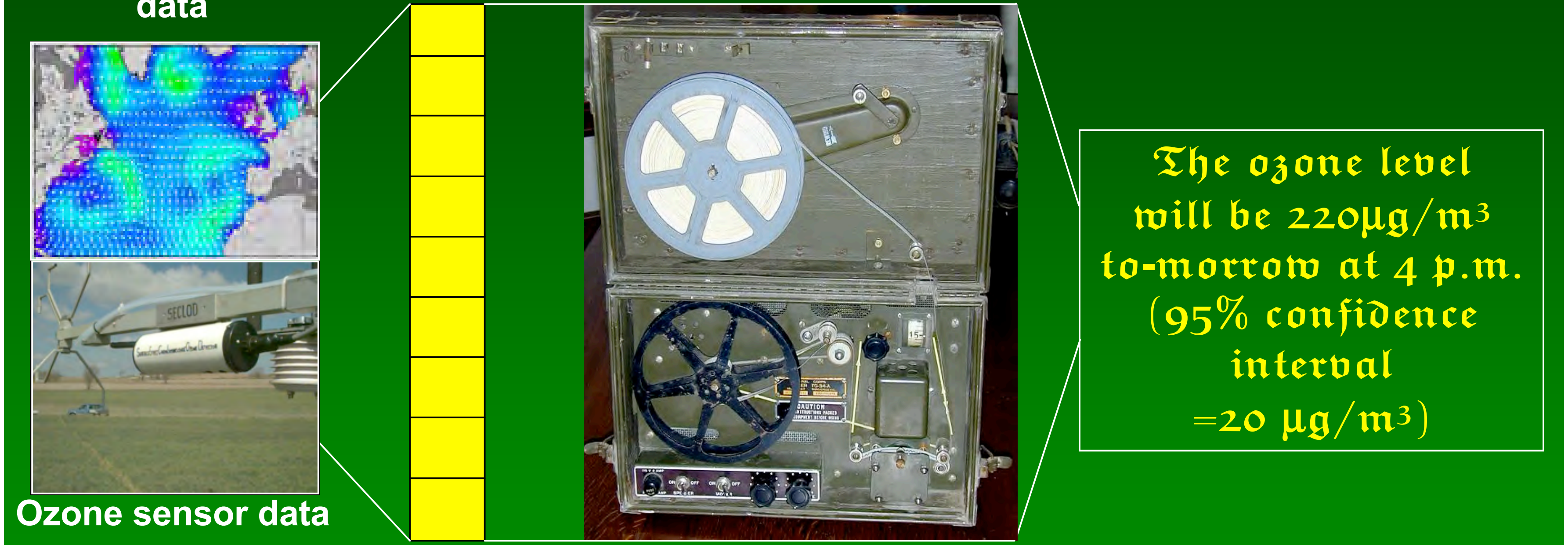
Les différentes applications

En principe, les réseaux de neurones s'utilisent dans tous les domaines où l'on retrouve des phénomènes non linéaires : physique, chimie, biologie, mécanique, astronomie

CONVENTIONAL MACHINE LEARNING

Feature vector

Weather forecast data



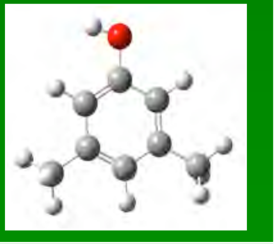
Ozone sensor data

The ozone level will be 220µg/m^3 to-morrow at 4 p.m. (95% confidence interval =20 µg/m^3)

« Learning machine »
(linear, polynomial, neural network,
kernel machine, support vector machine, …)

UNCONVENTIONAL MACHINE LEARNING

What if we want to learn from STRUCTURED DATA ?



Shift from *vector machines*
to
graph machines

WHAT IS “TRAINING”?

Training is an algorithmic process whereby the parameters of the model are estimated in order to minimize the discrepancy between the experimental target values and the corresponding predicted values.

Typically, a minimum of the *least-squares cost function* with respect to the parameters of the model is sought

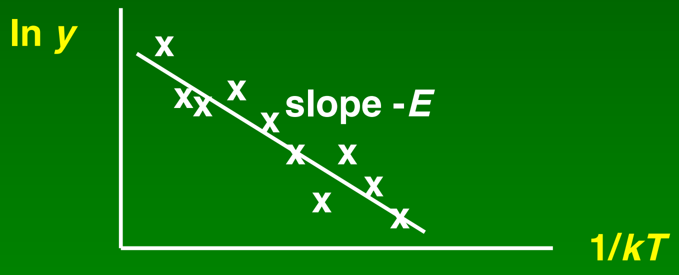
$$J(\theta) = \sum_{k=1}^{N} \left[y_p^k - g_\theta\left(\mathbf{x}^k\right) \right]^2$$

where $g_\theta(.)$ is the model, with parameter vector θ.

STATISTICAL MACHINE LEARNING *vs.* REGRESSION

- REGRESSION :
  - A knowledge-based model is available, and is considered to be “the truth”; it features unknown parameters.

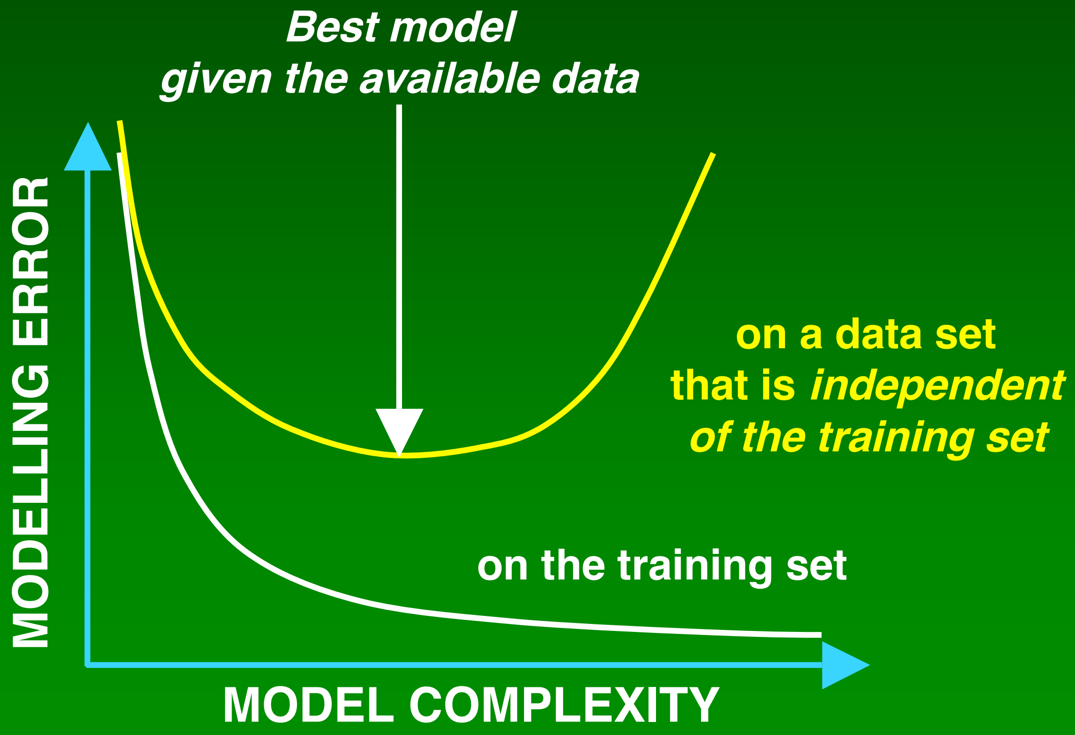
    Example : $y = A \exp(-E / kT)$
  - The parameters are estimated by statistical techniques (e.g. least squares fitting); confidence intervals for the parameters are estimated.



- MACHINE LEARNING :
  - No « true » model is available; a *predictive* model is sought, from the available data.
  - The ability of the model to *generalize* must be estimated.

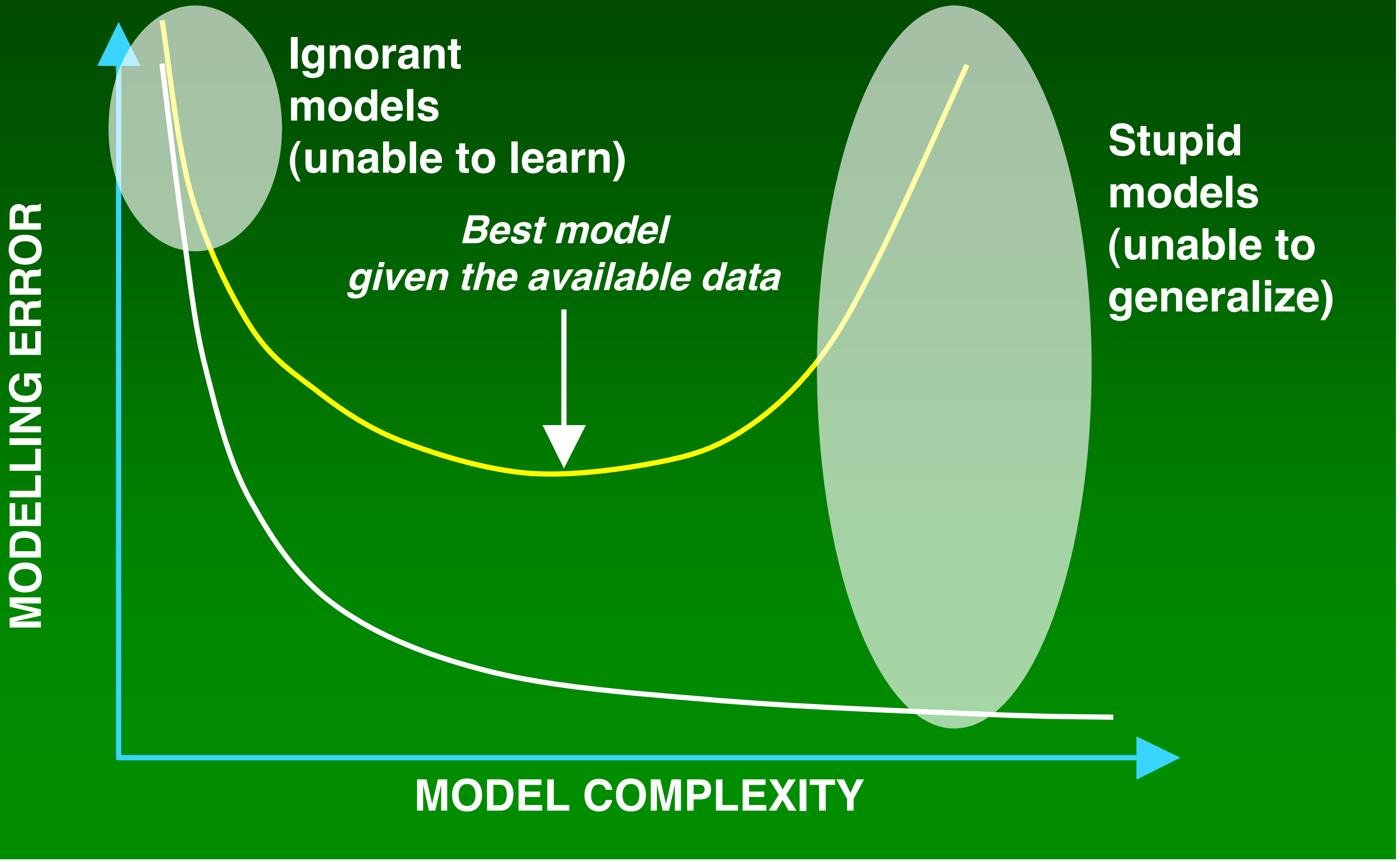
WHAT IS “GENERALIZATION” ?

Generalization is the ability of the model to provide satisfactory predictions for situations that are not present in the training set*.*



A.I. AS A TRADEOFF…

Artificial intelligence is a tradeoff between ignorance and stupidity…



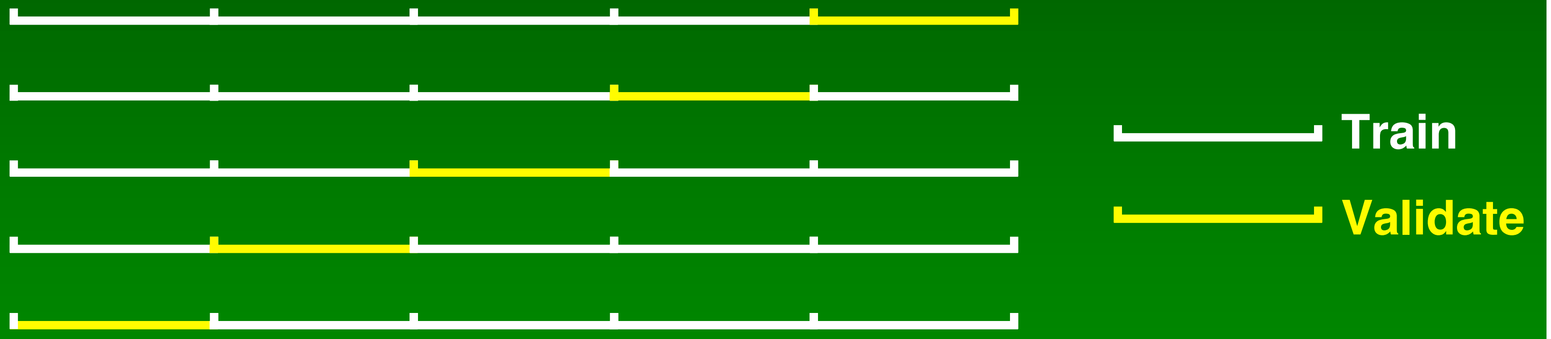
HOW TO ESTIMATE
THE GENERALIZATION ABILITY OF A MODEL

- ***HOLD-OUT:***

  When data is plentiful, training is performed on a part of the available data, and the prediction error on the rest of the data is computed.

- ***CROSS-VALIDATION***

  Split the available data into *D* subsets. Perform (*D* = 5, *N* examples):



Compute *VRMSE* $= \sqrt{\dfrac{1}{N}\sum_{k=1}^{N}\left[y_p^k - g_\theta\left(\mathrm{x}^k\right)\right]^2}$ **where $g_\theta(\mathrm{x}^k)$ is the prediction performed by the model on example *k* when it is *in the validation set*.**

HOW TO ESTIMATE THE GENERALIZATION ABILITY OF A MODEL

LEAVE-ONE-OUT

Cross-validation with $D = N$: a single example is withdrawn from the training set, training is performed on all other examples.

Leave-one-out score:

$$LOO = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left[y_p^k - g_\theta^{-k}\left(\mathbf{x}^k\right)\right]^2} = \sqrt{\frac{1}{N}\left(R_k^{-k}\right)^2}$$

The leave-one-out score is an unbiased estimator of the generalization error.

Very computer-intensive!

HOW TO ESTIMATE THE GENERALIZATION ABILITY OF A MODEL

VIRTUAL LEAVE-ONE-OUT:

Train with *all* examples, and approximate the prediction error on example *k* if it had been withdrawn from the training set as

$$R_k^{-k} \approx \frac{R_k}{1-h_{kk}}$$

where h_{kk} is the *leverage* of example *k.*

Virtual leave-one-out score:

$$VLOO = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(\frac{R_k}{1-h_{kk}}\right)^2}$$

(exact for linear-in-their-parameters models, known as the PRESS statistic).

THE LEVERAGES

The leverages are the diagonal elements of the “hat matrix”

$$\mathbf{H} = \mathbf{Z}\left(\mathbf{Z}^{\mathrm{T}}\mathbf{Z}\right)^{-1}\mathbf{Z}^{\mathrm{T}}$$

where Z is the jacobian matrix of the model

$$\mathbf{Z} = \left[\frac{\partial g_\theta\left(\mathbf{x}^i\right)}{\partial \theta_1} \quad \frac{\partial g_\theta\left(\mathbf{x}^i\right)}{\partial \theta_2} \quad \ldots \quad \frac{\partial g_\theta\left(\mathbf{x}^i\right)}{\partial \theta_p} \right]$$

(*N* x *p* matrix where *N* is the number of examples and *p* is the number of parameters)

THE LEVERAGES

- Since the leverages are the diagonal elements of an orthogonal projection matrix, they have the following properties:

$$\sum_{k=1}^{N} h_{kk} = p \text{ (number of parameters)}$$

$$0 < h_{kk} < 1 \quad \forall k$$

- INTERPRETATION:

h_{kk} is the proportion of the parameters of the model that is used for fitting the model to observation k,

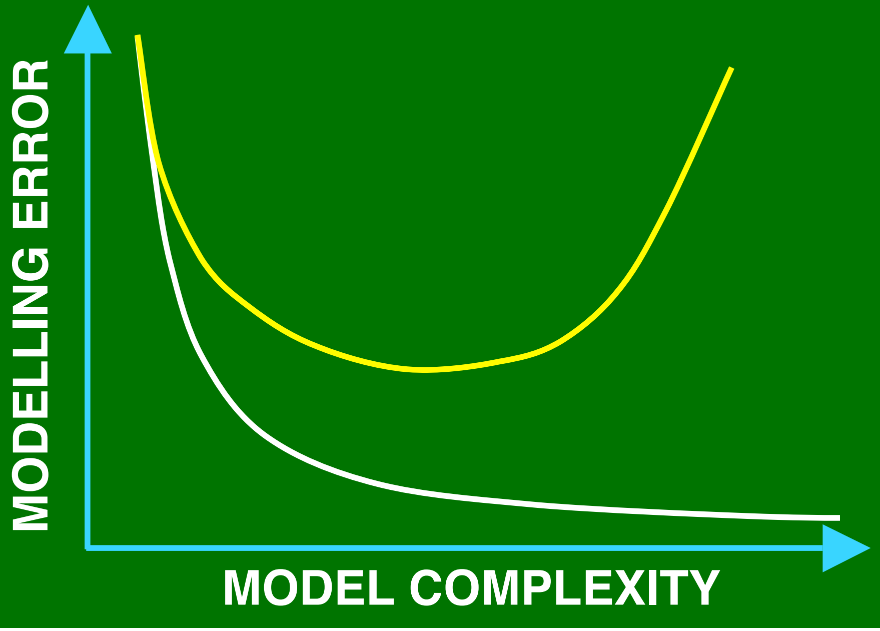
HENCE A MODEL THAT HAS LARGE LEVERAGES
IS VERY LIKELY TO EXHIBIT OVERFITTING

A MODEL SELECTION STRATEGY

- TRAIN MODELS OF INCREASING COMPLEXITY

  e.g. polynomials of increasing degree, neural networks with increasing number of hidden neurons, …

- STOP WHEN THE ESTIMATED GENERALIZATION ERROR STARTS INCREASING.



OUTLINE

CONVENTIONAL MACHINE LEARNING: AN OVERVIEW

LEARNING NUMBERS FROM GRAPHS: GRAPH MACHINES

MODEL SELECTION FOR GRAPH MACHINES: VIRTUAL LEAVE-ONE-OUT

APPLICATION TO COMPUTER-AIDED DRUG DESIGN

UNCONVENTIONAL MACHINE LEARNING: LEARNING FROM GRAPHS

- ***PURPOSE:*** learn a mapping from a set of graphs to a “target” set of real numbers (*regression*) or of binary numbers (*classification*).

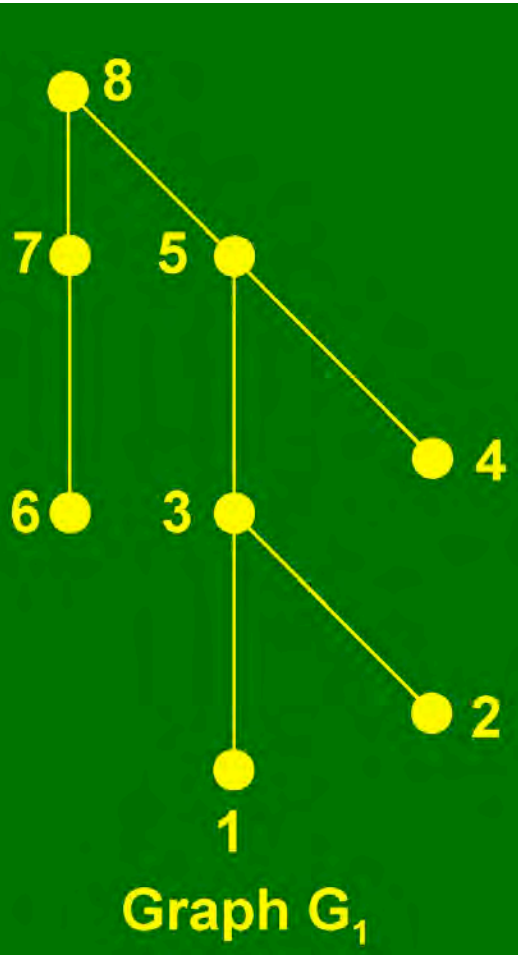
- *PHILOSOPHY:*
  - find a vector representation of each graph;
  - map the set of representations to the target set.

Do both simultaneously!

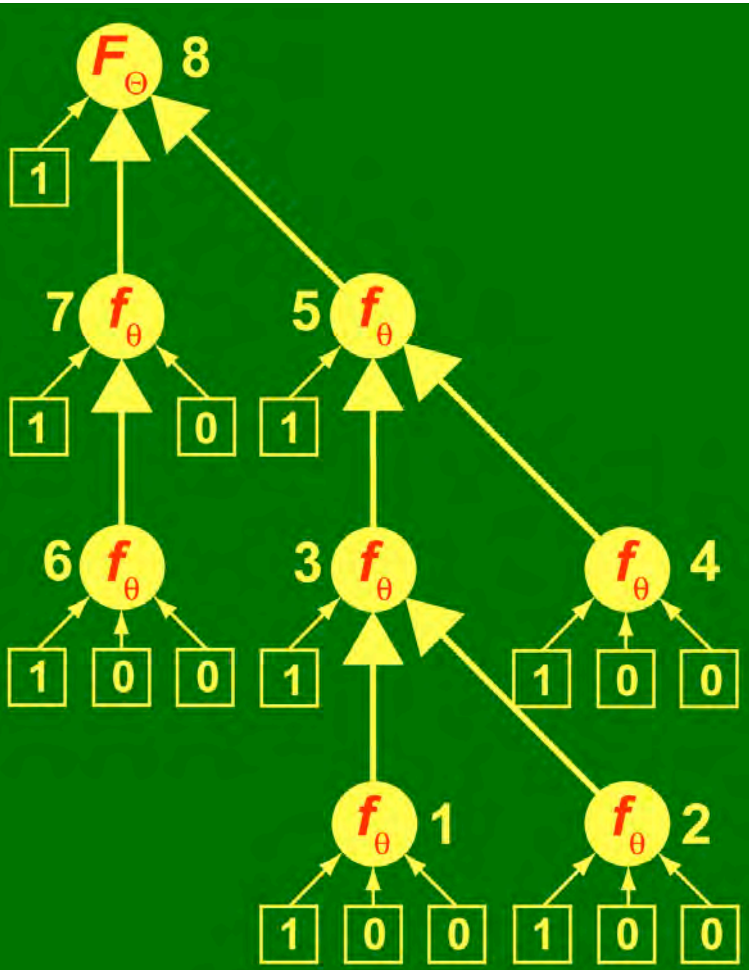
- Combination of two simple principles:
  - if there is some structure in the data to learn from, build the structure into the learning machine (“semi-physical modelling”, Y. Oussar & G. Dreyfus, 2001);
  - if you can’t handcraft a representation, learn it (“convolutional neural networks”, LeCun et al., 1989).

- Reminiscent of Labeled Recursive Auto-Associative Memories (Sperduti, 1994).

DESIGN OF A GRAPH MACHINE



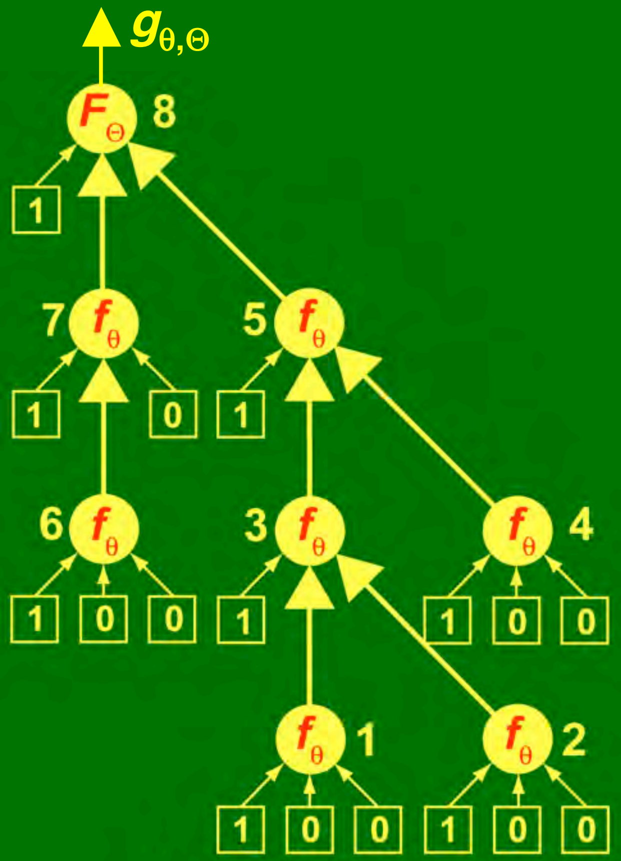
Graph G_1



Graph machine

DESIGN OF A GRAPH MACHINE

$$g^1_{\theta,\Theta}\left(x_1, x_2, \ldots, x_8\right) = F_\Theta\left(1, f_\theta\left(1, f_\theta\left(1,0,0\right),0\right), f_\theta\left(1, f_\theta\left(1, f_\theta\left(1,0,0\right), f_\theta\left(1,0,0\right)\right), f_\theta\left(1,0,0\right)\right)\right)$$

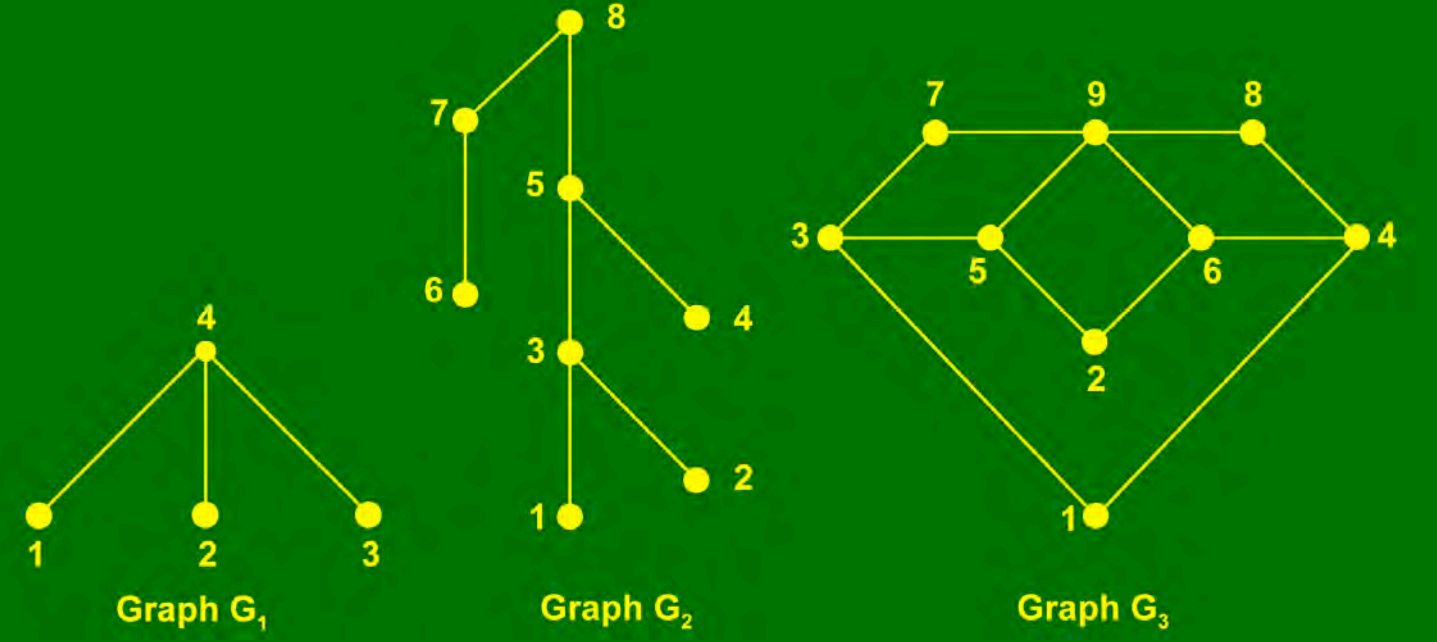


Graph machine

If f_θ is a neural network:
"recursive network"
(Frasconi et al., 1998)

TWO DIDACTIC EXAMPLES: LEARNING HOW TO COUNT THE NODES AND EDGES OF GRAPHS

STEP 1: TRAINING SET AND TARGET VALUES



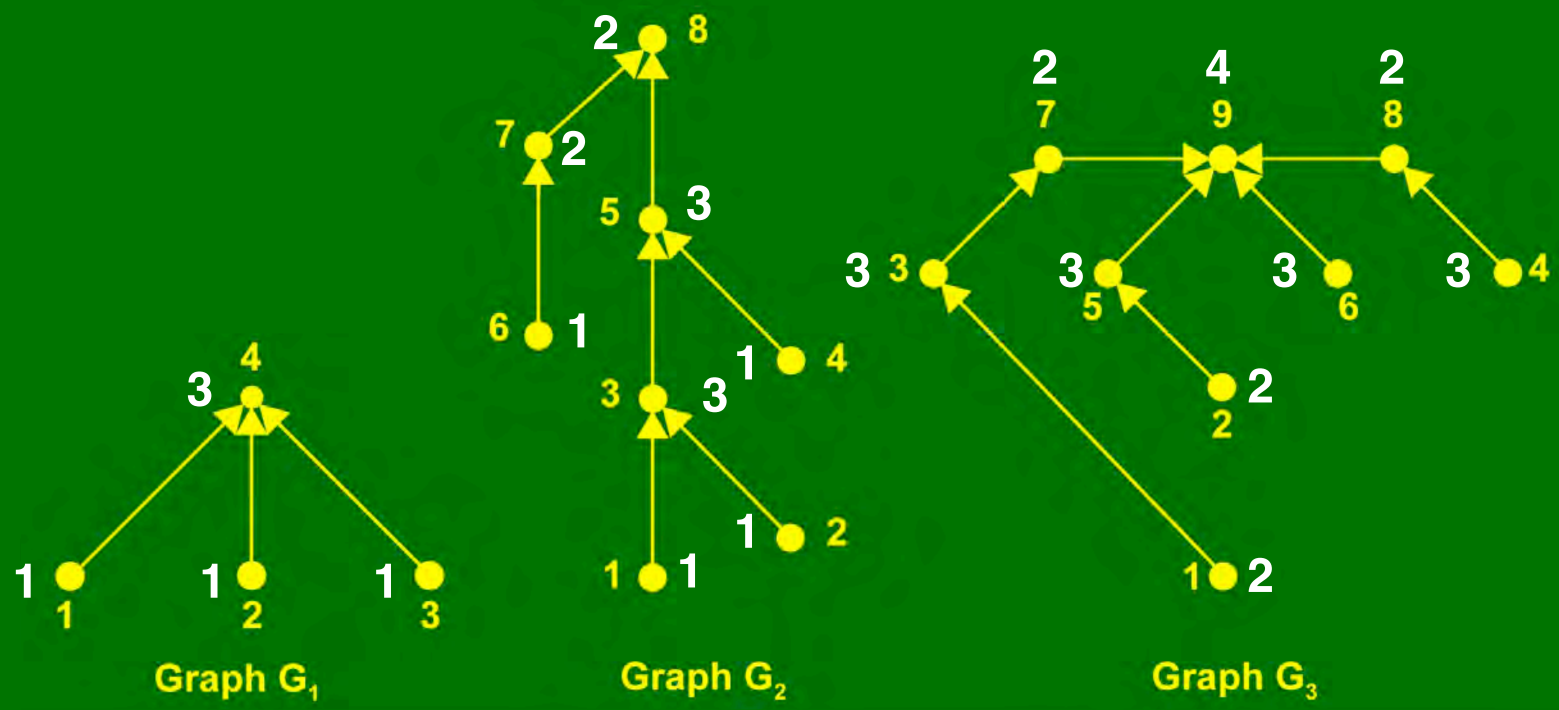
Target values:

Nodes	4	8	9
Edges	3	7	12

TWO DIDACTIC EXAMPLES: LEARNING HOW TO COUNT THE NODES AND EDGES OF GRAPHS

STEP 2: TURN THE GRAPHS INTO DIRECTED ACYCLIC GRAPHS

Assign a label to each node: its *degree* in the original graph



TWO DIDACTIC EXAMPLES:
LEARNING HOW TO COUNT THE NODES AND EDGES OF GRAPHS

STEP 3: POSTULATE A FAMILY OF NODE FUNCTIONS

e. g. affine, polynomial, neural network, …

STEP 4: TRAIN THE MACHINES

by minimizing the cost function *J* with respect to the parameters

$$J(\theta) = \sum_{k=1}^{N} \left[y_p^k - g_\theta^k \right]^2$$

Notice the difference with the conventional cost function

$$J(\theta) = \sum_{k=1}^{N} \left[y_p^k - g_\theta\left(\mathbf{x}^k\right) \right]^2 \quad !$$

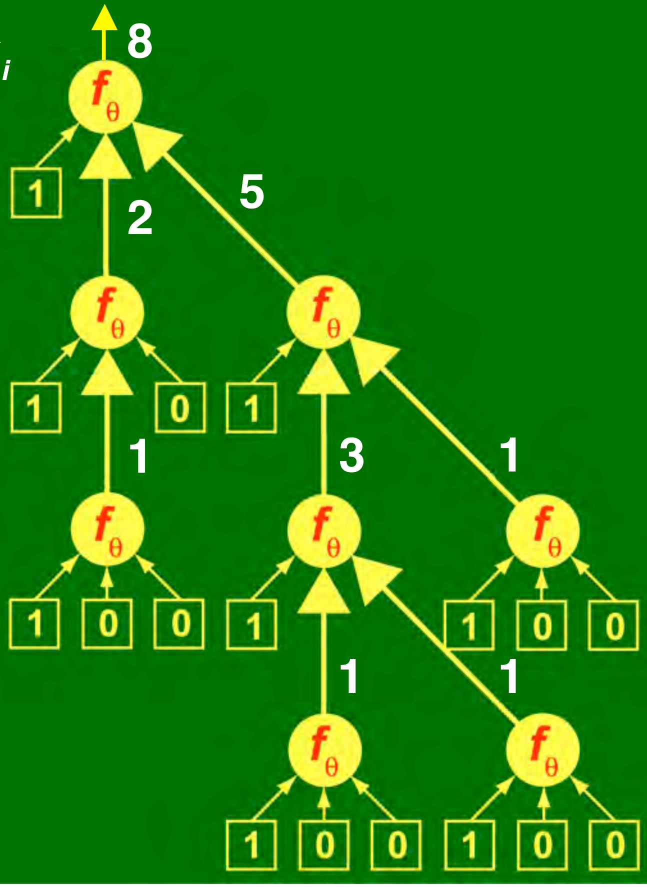
TWO DIDACTIC EXAMPLES: LEARNING HOW TO COUNT THE NODES AND EDGES OF GRAPHS

HOW TO COUNT THE NODES OF A GRAPH

Postulated node function : affine function

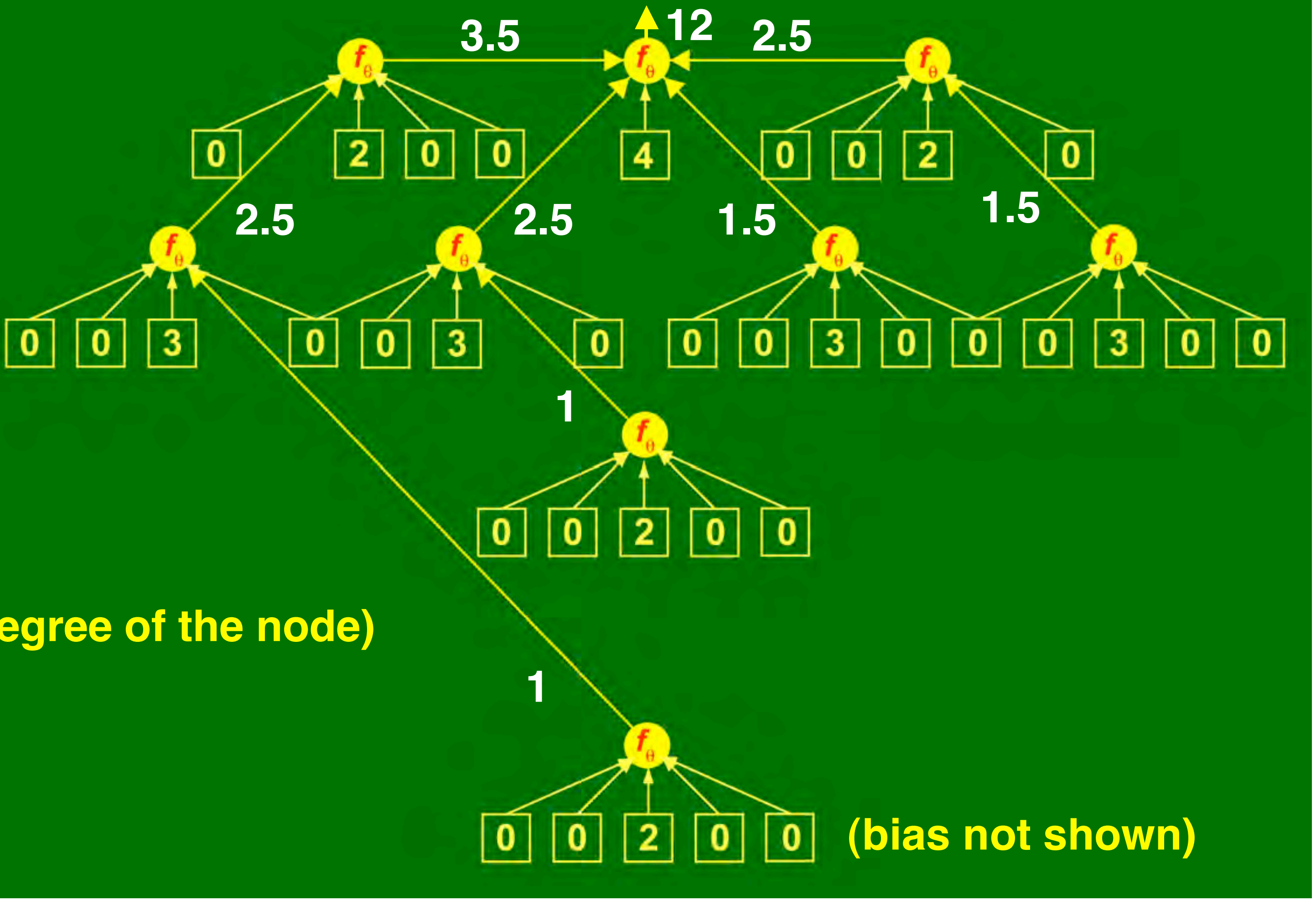
$$f_\theta(\mathbf{x}) = \sum_i \theta_i x_i$$

Solution: $f_\theta(\mathbf{x}) = \sum_i x_i$



TWO DIDACTIC EXAMPLES: LEARNING HOW TO COUNT THE NODES AND EDGES OF GRAPHS

HOW TO COUNT THE EDGES OF A GRAPH



Solution:

$\theta_0 = 0$ (bias)

$\theta_1 = \theta_2 = 1$

$\theta_3 = 1/2$ ($x_3 =$ degree of the node)

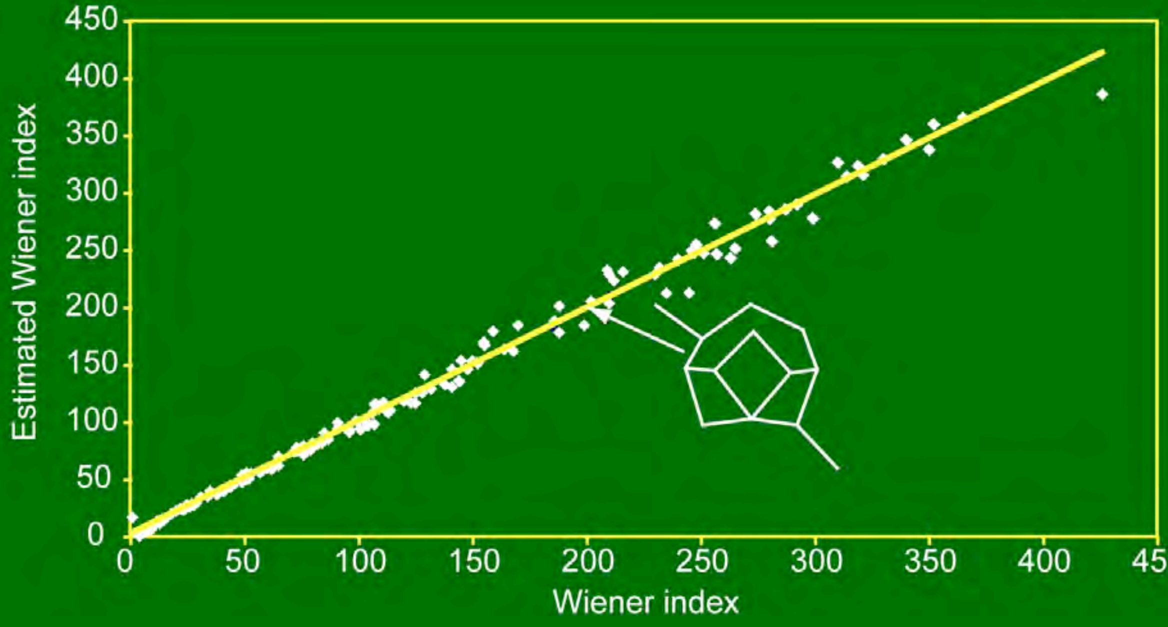
$\theta_4 = \theta_5 = 1$

(bias not shown)

A NONLINEAR EXAMPLE: LEARNING THE WIENER INDEX

WIENER INDEX OF A GRAPH: SUM OF THE DISTANCES BETWEEN ITS NODES

Distance between two nodes = number of edges in the shortest path between the nodes



OUTLINE

CONVENTIONAL MACHINE LEARNING: AN OVERVIEW

LEARNING NUMBERS FROM GRAPHS: GRAPH MACHINES

MODEL SELECTION FOR GRAPH MACHINES: VIRTUAL LEAVE-ONE-OUT

APPLICATION TO COMPUTER-AIDED DRUG DESIGN

MODEL SELECTION FOR GRAPH MACHINES

- CONVENTIONAL MACHINE LEARNING:

$$R_k^{-k} \approx \frac{R_k}{1 - h_{kk}}$$

h_{kk} (the leverage of observation *k*) is the diagonal element of the hat matrix $\mathrm{H} = \mathrm{Z}\left(\mathrm{Z}^{\mathrm{T}}\mathrm{Z}\right)^{-1}\mathrm{Z}^{\mathrm{T}}$

where Z is the Jacobian matrix of the model

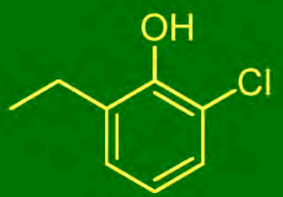
$$z_{ij} = \frac{\partial g_\theta\left(\mathbf{x}^i\right)}{\partial \theta_j}$$

- GRAPH MACHINES: the same results hold true, but, instead of the Jacobian matrix, one has $z_{ij} = \frac{\partial g_\theta^i}{\partial \theta_j}$

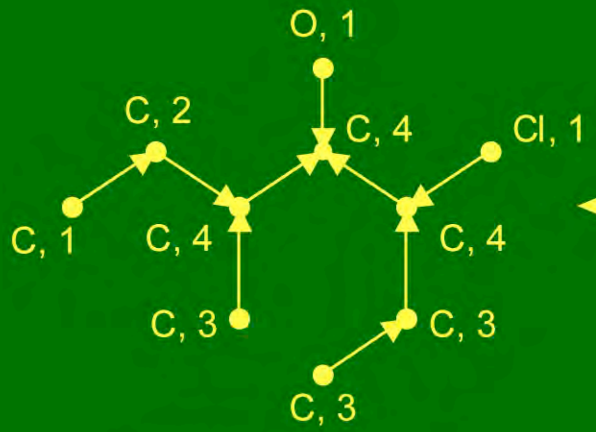
ENCODING MOLECULES AS GRAPHS

OC1=C(Cl)C=CC=C1CC

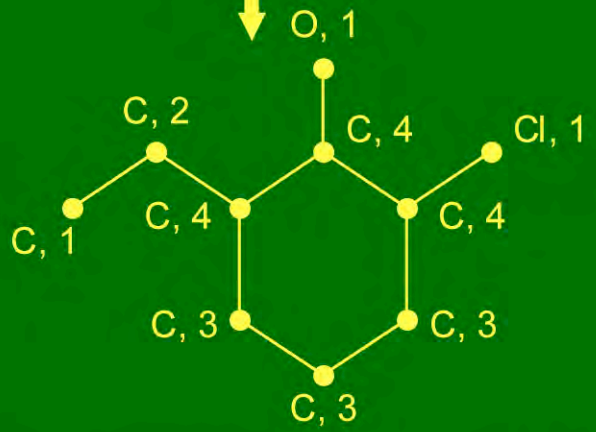
SMILES representation



Molecule



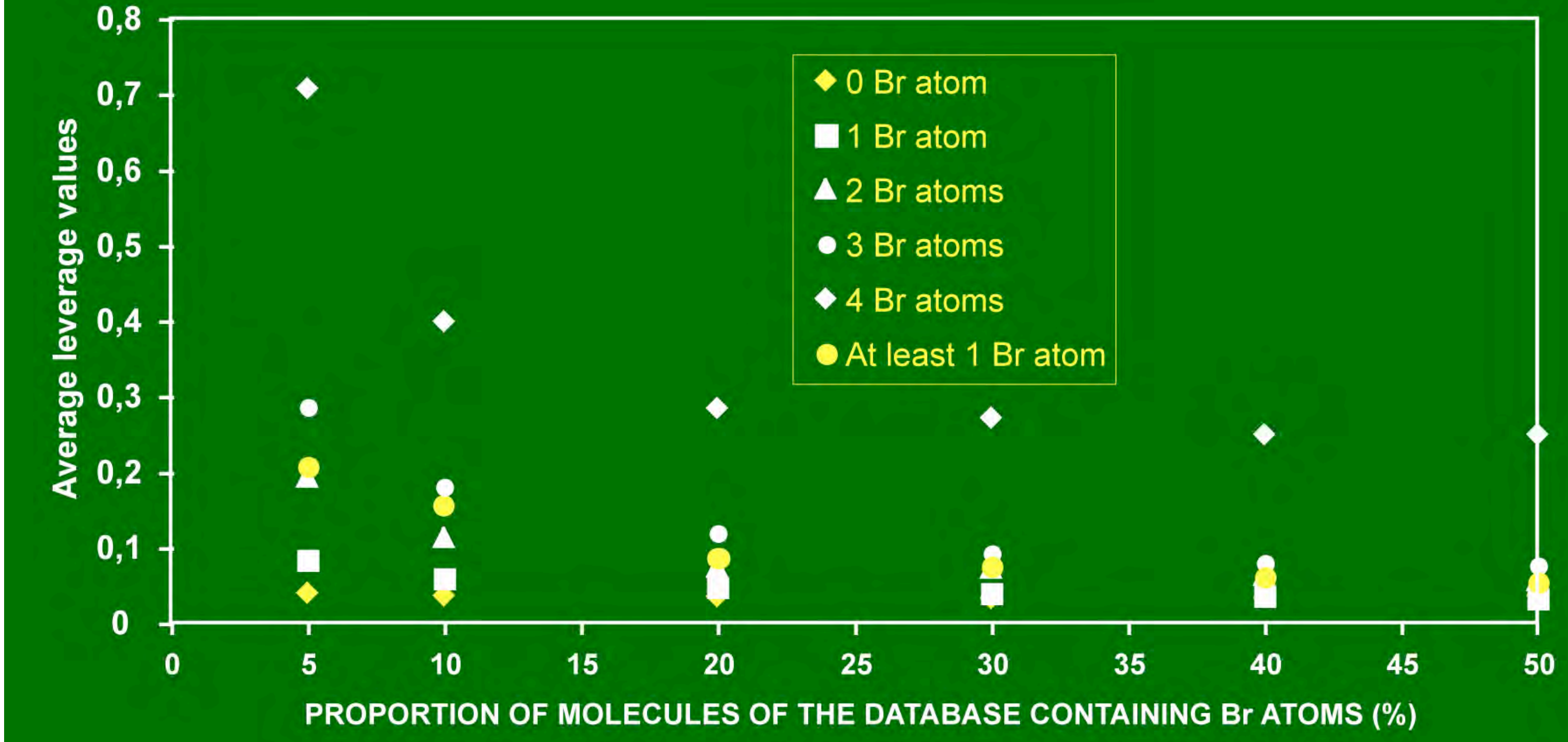
Directed acyclic graph



Graph representation (undirected and cyclic)

EXAMPLE: LEARNING THE MASS OF MOLECULES

- 6 training sets, 330 molecules each, involving C, H, F, Br, Cl atoms
- The proportion of molecules containing Br atoms varies from 5% to 50%

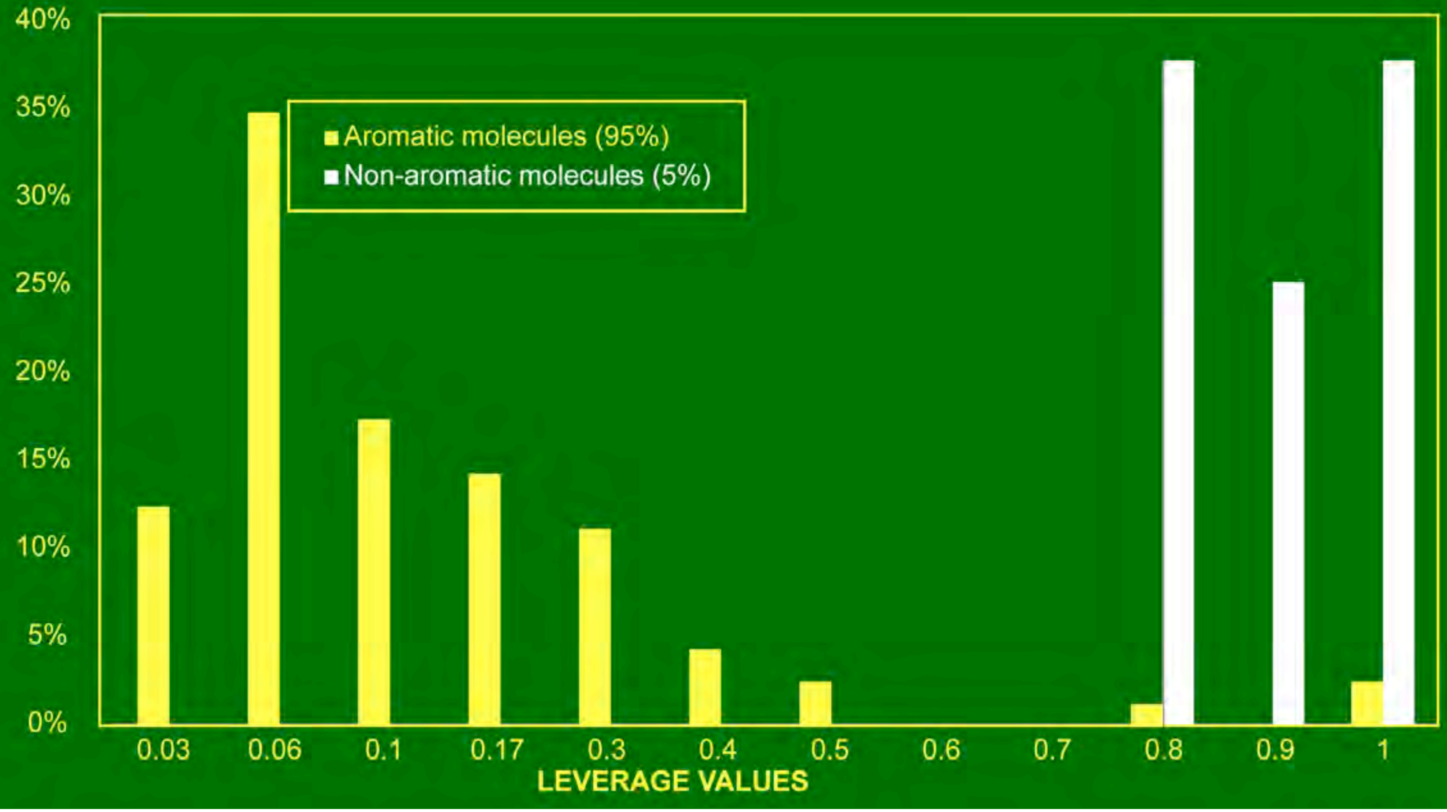


EXAMPLE: DISCRIMINATING AROMATIC FROM NON-AROMATIC MOLECULES (1)

6 training sets, 170 molecules each.

Histogram of the leverages

when the proportion of non-aromatic molecules in the training set is 5%

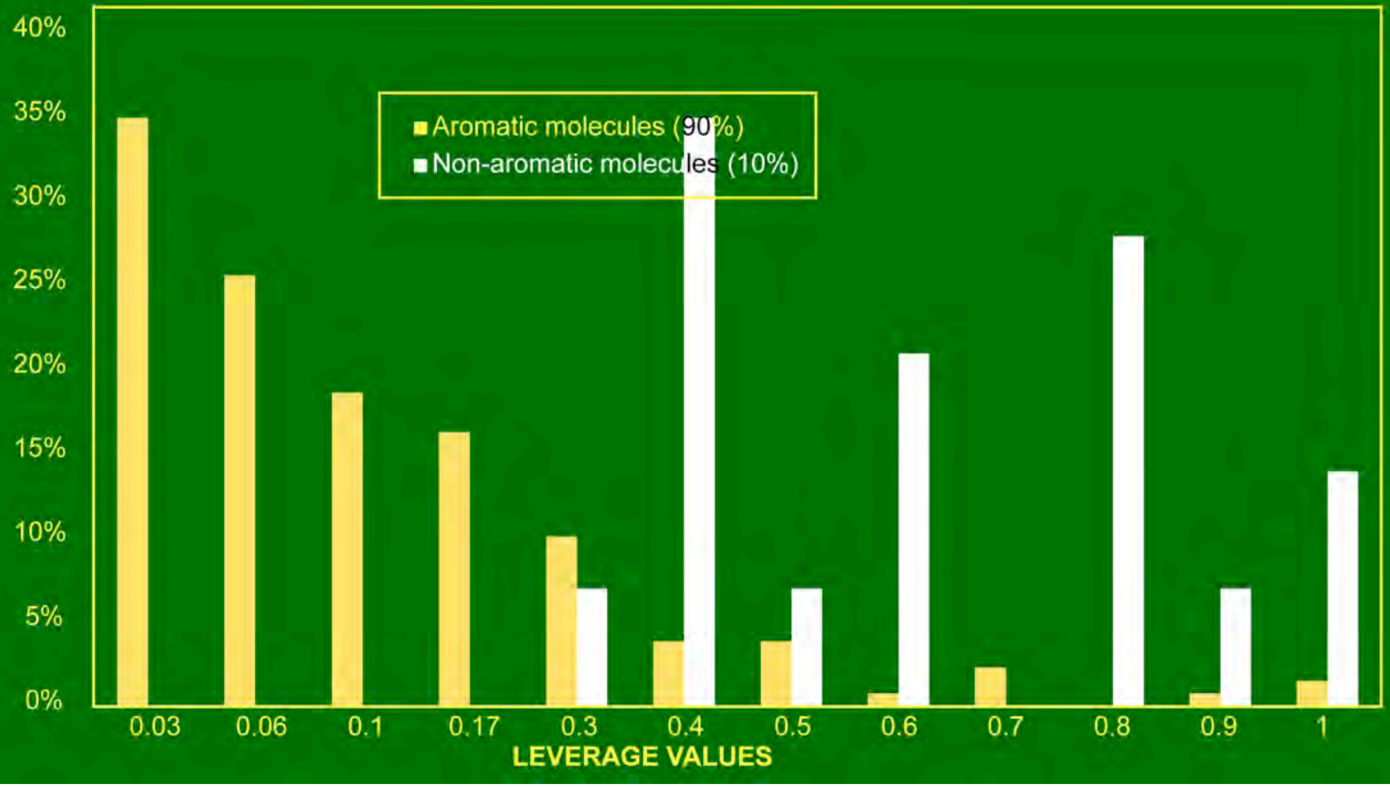


EXAMPLE: DISCRIMINATING AROMATIC FROM NON-AROMATIC MOLECULES (2)

6 training sets, 170 molecules each.

Histogram of the leverages

when the proportion of non-aromatic molecules in the training set is 10%

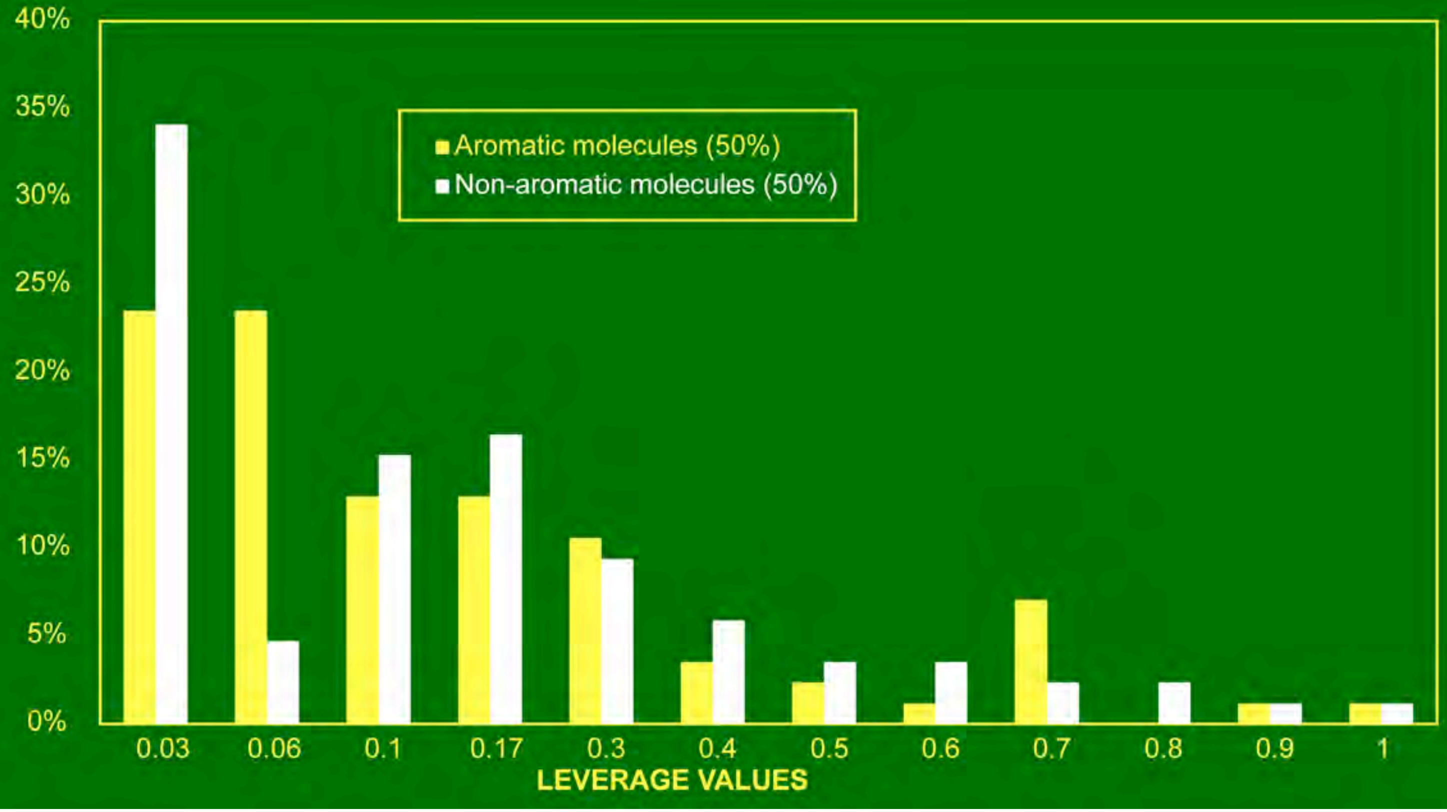


EXAMPLE: DISCRIMINATING AROMATIC FROM NON-AROMATIC MOLECULES (3)

6 training sets, 170 molecules each.

Histogram of the leverages

when the proportion of non-aromatic molecules in the training set is 50%



OUTLINE

CONVENTIONAL MACHINE LEARNING: AN OVERVIEW

LEARNING NUMBERS FROM GRAPHS: GRAPH MACHINES

MODEL SELECTION FOR GRAPH MACHINES: VIRTUAL LEAVE-ONE-OUT

APPLICATION TO COMPUTER-AIDED DRUG DESIGN

THE RATIONALE FOR QSAR/QSPR
Quantitative Structure-Activity/Property Relationships

THE UNIVERSE: 10^{22} stars

KNOWN DRUGS: 2,000 molecules

KNOWN CHEMICALS: 22 10^{6}

ESTIMATED NUMBER OF MOLECULES: 10^{60}

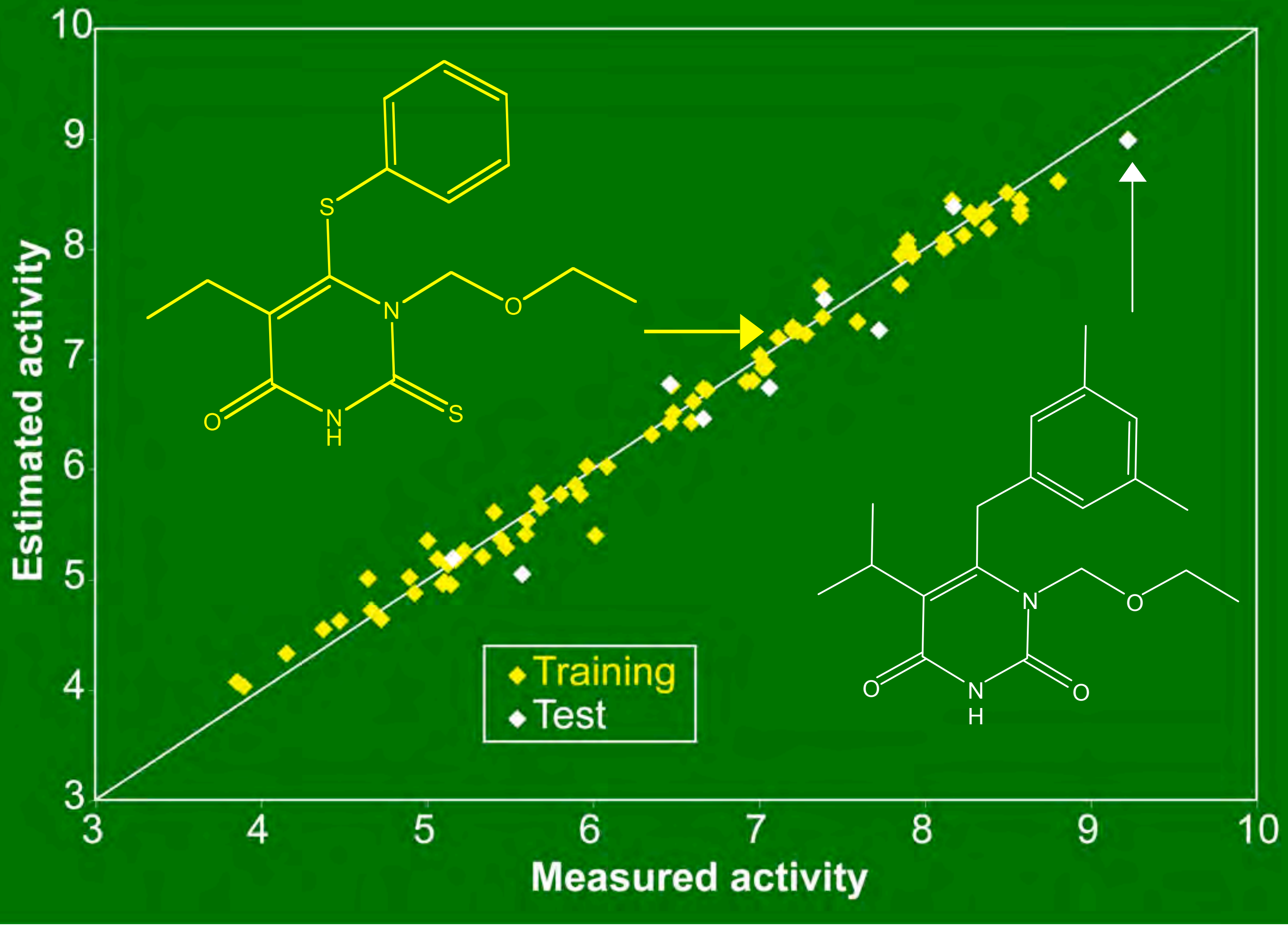
(source: Pierre Baldi)

THE CONVENTIONAL APPROACH TO QSAR/QSPR

- FIND AN APPROPRIATE SET OF VARIABLES (« DESCRIPTORS ») THAT HAVE AN INFLUENCE ON THE QUANTITY TO BE PREDICTED.
- MEASURE OR COMPUTE THE DESCRIPTORS. **VERY COSTLY!**
- CHECK THEIR ACTUAL RELEVANCE (« VARIABLE SELECTION »)
- INPUT THEM TO THE POSTULATED MODEL (LINEAR, POLYNOMIAL, NEURAL NETWORK, KERNEL MACHINE, SVM, …)

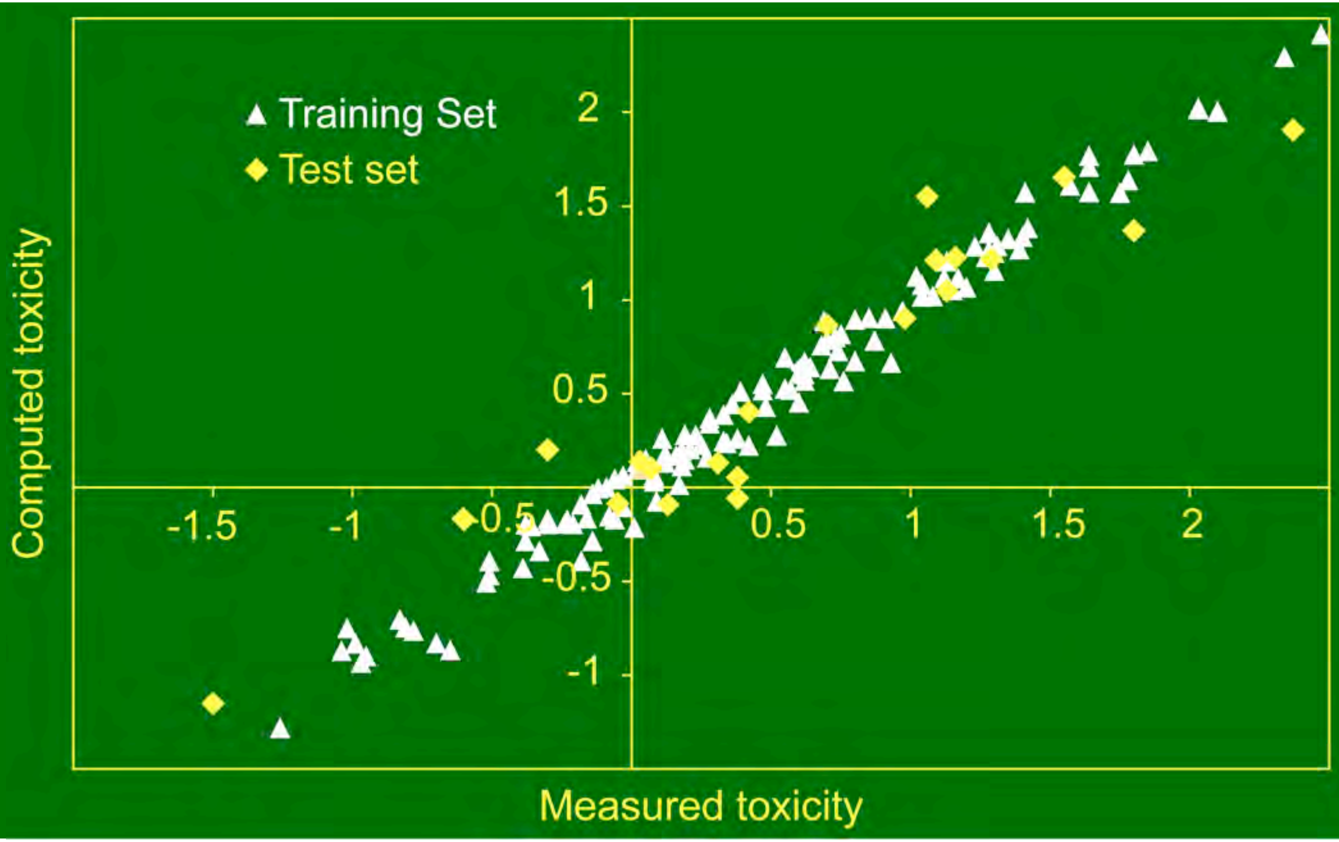
GRAPH MACHINES EXEMPT THE MODEL DESIGNER FROM PERFORMING THE FIRST THREE STEPS!

GRAPH MACHINE PREDICTION OF ANTI-HIV PROPERTIES



GRAPH MACHINES *vs.* CONVENTIONAL QSAR/QSPR PREDICTION OF THE TOXICITY OF PHENOLS

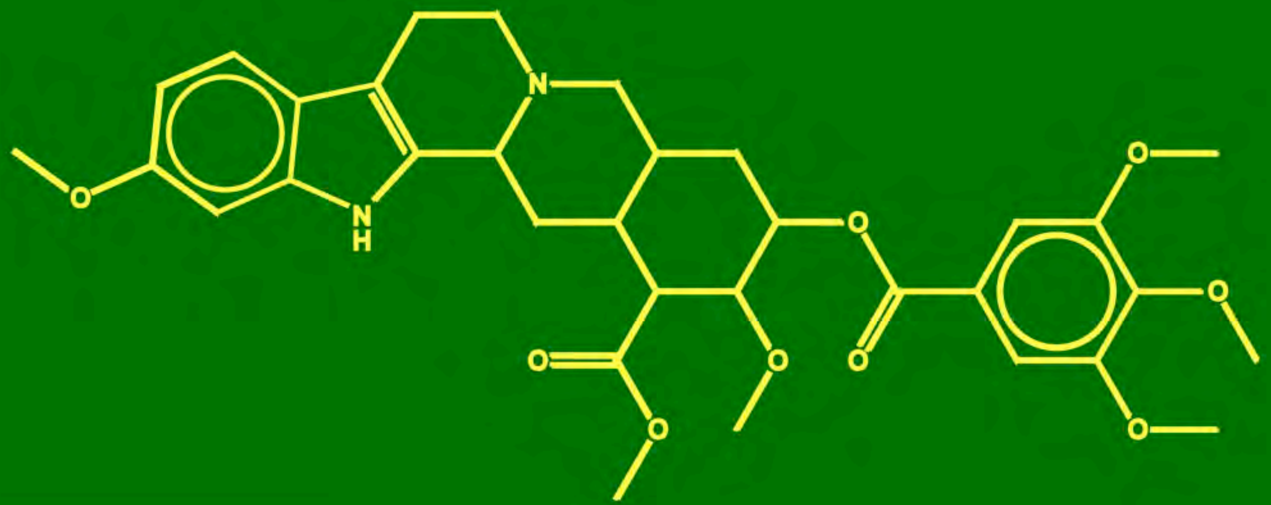
	GM	MLR	RBFNN	SVM
RMS ERROR (TEST SET)	0.27	0.46	0.29	0.35



GRAPH MACHINES *vs.* CONVENTIONAL QSAR/QSPR CLASSIFICATION OF COMPOUNDS: CARCINOGENIC/NON-CARCINOGENIC

306 MOLECULES, POTENTIALLY CARCINOGENIC TO FEMALE RATS

METHOD	ACCURACY (%)
GRAPH MACHINES	71
GRAPH KERNELS	67



SUMMARY: CONVENTIONAL *vs.* UNCONVENTIONAL MACHINE LEARNING

	CONVENTIONAL	UNCONVENTIONAL
Input	Vector of features	Graph structure
Design	1 machine for *N* examples	*N* machines for *N* examples
Training	1 output per example	Each machine is trained with a single example (shared weights)
Result	Vector-output mapping	Structure-output mapping

CONCLUSION

- EFFICIENT METHOD FOR LEARNING FROM STRUCTURED DATA WITHOUT HAVING TO COMPUTE SPECIFIC FEATURES FOR EACH SPECIFIC PROBLEM.

- *GOOD NEWS:* VIRTUAL LEAVE-ONE-OUT CAN BE EXTENDED TO GRAPH MACHINES.

- MANY OPEN PROBLEMS: EXPERIMENTAL PLANNING, ...

- *BAD NEWS:* 1 nsec computation time /molecule, 10^{40} molecules $\Rightarrow$ 10^{26} years…

FOR MORE INFORMATION

- Goulon, A., Picot, T., Duprat, A., Dreyfus, G.: *Predicting Activities without Computing Descriptors: Graph Machines for QSAR.* SAR and QSAR in Environmental Research, to be published.

- Goulon-Sigwalt-Abram, A., Duprat, A., Dreyfus, G.: *From Hopfield Nets to Recursive Networks to Graph Machines: Numerical Machine Learning for Structured Data*. Theoretical Computer Science 344 (2005) 298-334.

- Goulon-Sigwalt-Abram A., Duprat A., Dreyfus G.: *Learning numbers from graphs.* Applied Statistical Modeling and Data Analysis (2005). Available from http://www.neurones.espci.fr/Articles_PS/ASMDA.pdf

- Goulon-Sigwalt-Abram A., Duprat A., Dreyfus G.: *Graph Machines and their Applications to Computer-aided Drug Design: a New Approach to Learning from Structured Data.* Unconventional Computation 2006, Lecture Notes in Computer Science vol. 4135, 1 - 19 (Springer, 2006).

RECENT TEXTBOOK

A coherent and comprehensive, yet not redundant, practically-oriented introduction, written by experts and seemlessly edited.

495 pages.

SPRINGER, 2005

http://www.springer.com/east/home/generic/search/results?SGWID=5-40109-22-34174366-0