Available online at www.sciencedirect.com
ScienceDirect
Catalysis Today 137 (2008) 2–10
CATALYSIS TODAY
www.elsevier.com/locate/cattod

Data mining in catalysis: Separating knowledge from garbage

Gadi Rothenberg *

Van't Hoff Institute for Molecular Sciences, University of Amsterdam, Nieuwe Achtergracht 166, 1018 WV Amsterdam, The Netherlands

Available online 28 March 2008

Abstract

The subject of 'data mining', also known as 'knowledge discovery', has exciting applications in catalysis research. In this paper, I outline the basics of data mining, with examples from different areas of catalysis. The focus is on the concepts, rather than on specific algorithms. I highlight the subject of descriptor modelling methods for rational catalyst design, and summarise some simple validation approaches that can help us separate knowledge from garbage.

Keywords: Knowledge discovery; Predictive modelling; Statistical analysis; High-throughput experimentation; Model validation; Virtual library

1. Introduction

This paper gives my personal overview on the exciting subject of data mining in catalysis [1]. It follows the plenary lecture that I gave at the 2007 EuroCombicat conference in Bari, Italy, and (with the editor's kind consent) I have kept the narrative of the lecture. The paper thus focuses on the basic concepts and the possibilities of data mining, rather than on specific algorithms mathematical techniques. That said, I include several references to articles containing introductory tutorials on the various methods described here. Note that although data mining is sometimes considered as a branch of statistics, it may also employ a variety of other techniques. Practically, data mining is usually performed on very large datasets extracted from databases, and thus sampling methodologies are of prime importance [2,3].

The first thing we must understand and accept about computer models is that they are very similar to experiments. Unless they are trivially simple, you cannot predict their results just from looking at the code. Moreover, if they are planned badly, or if they are programmed badly, they may either crash or yield meaningless numbers (the computer equivalent of brown goo). Just like experiments, computer modelling is hardware-dependent, and yet, just like in experiments, you can sometimes obtain surprisingly nice results using relatively simple equipment. Importantly, computer models offer us no understanding, only numbers. We must examine the meaning and the worthiness of these numbers, always considering also the statistical errors (e.g. noise in measurements and sampling influence) involved. Unfortunately, too many scientists tend to accept the results of "successful" computer models at face value. Just because a program did not crash does not mean that the results are meaningful!

One important area where computers play an essential role is data analysis and data mining, particularly in analysing large sets of reactions from combinatorial experiments. Statistical methods such as principal component analysis (PCA), partial least squares (PLS), and artificial neural networks (ANNs), can highlight trends in large datasets. Knowing these trends, and the key parameters that govern them, often leads to a deeper understanding of the catalytic cycle.

2. The basics of data mining

High-throughput experimentation (HTE) has changed the way we do catalysis research. Robotic systems can now perform hundreds of experiments per day. This yields mind-boggling amounts of experimental results. HTE can help us screen larger regions of the catalyst space. But, the total space is much too large for exhaustive screening, even using robots, so we must choose which areas to search in. Moreover, although HTE gives overwhelming quantities of data, much of it is 'garbage data' that must be sifted out [4]. In this section, I

* Fax: +31 20 525 5604.
E-mail address: gadi@science.uva.nl.
URL: http://www.science.uva.nl/~gadi

0920-5861/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.cattod.2008.02.014

describe a few useful methods for filtering out garbage data, with some examples of their applications in catalysis research. For more details see the books of Massart et al. [5] and Tranter [6].

Fig. 1 shows a simplified data analysis flowchart. First, the data is collected and treated in a pre-processing step. Depending on the situation and on the knowledge available, you may want to mean-centre and/or scale the entire dataset, or perhaps divide it into subsets according to chemical restrictions. In many cases, including chemical knowledge and reaction restrictions will simplify the system. Then, dimension reduction methods such as PCA are used for highlighting the number of key variables, and linear/nonlinear regression models are built, tested, and validated. The model's predictions can be fed back to the data collection stage, improving the figures of merit in an iterative fashion.

The simplest form of regression is multiple linear regression (MLR), $\mathbf{Y} = \mathbf{XB} + \mathbf{E}$. Here, $\mathbf{X}$ contains the descriptors, $[d_1, \ldots, d_n]$, $\mathbf{B}$ contains the regression coefficients, $\mathbf{Y}$ contains the figures of merit $\mathbf{Y}$, and $\mathbf{E}$ contains the residuals (here, matrices are noted by bold capitals and vectors by bold lowercase letters). One well-known example of MLR is the relationship shown in Eq. (1). This model requires a few well characterized parameters $d_1, \ldots, d_n$, that are usually derived from experimental measurements or from quantum mechanical (QM) calculations. There are several applications of MLR in catalysis, e.g. the quantitative analysis of ligand effects (QALE) model, developed by Fernandez et al. [7].

$$\ln k = \mathbf{a}d_1 + \mathbf{b}d_2 + \cdots + \mathbf{z}d_n + \mathbf{E} \tag{1}$$

The trouble is that you often have too many descriptors, and/or insufficient information on the reaction mechanism. This creates two problems: building a regression model requires the calculation of the inverse of $\mathbf{X}^\mathbf{T}\mathbf{X}$, which cannot be done for a matrix $\mathbf{X}$ that contains more variables than experiments. Moreover, if you have too many descriptors, you can always find a so-called chance model that fits your data perfectly, but has no statistical relevance. In such cases, you must find the right descriptors and the right way of correlating them to the figures of merit (this is much more difficult when dealing with solids rather than with molecules). One approach is first to reduce the number of parameters using PCA, and then choose the analysis method. You can use either linear or non-linear models (Fig. 1). Both approaches are equally valid. Linear models, such as PLS regression, are more robust, and easier to interpret. Non-linear methods, such as ANNs, can handle more complicated systems, but they are "black box" models.

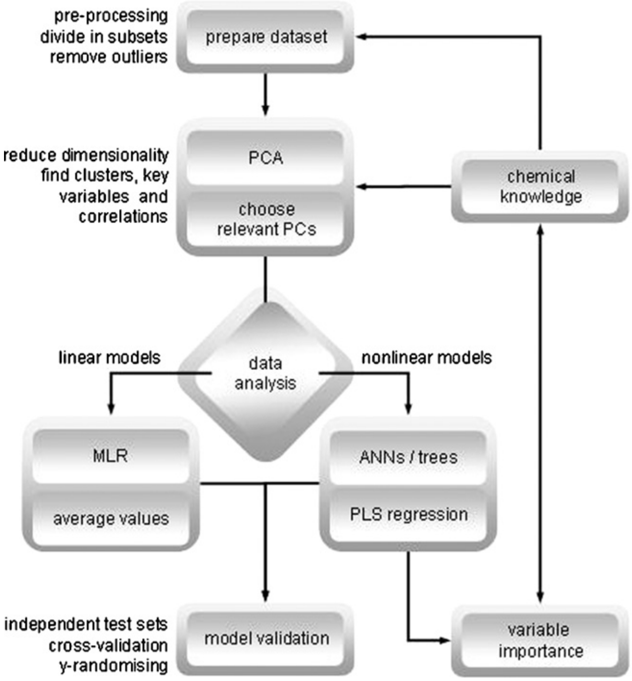

Fig. 1. Flowchart showing a general approach to data mining.

Selecting the right variables often improves the models and makes interpretation easier. When there are too many descriptors, and especially when these descriptors do not have a clear physico-chemical meaning (e.g. connectivity indices and other 2D descriptors), stochastic methods such as genetic algorithms and evolutionary strategies can be used for finding an optimal subset of descriptors [8,9].

2.1. Principal components analysis (PCA)

Suppose that you have an experimental data matrix that contains the concentration profiles of 12,500 reactions, performed using 50 different catalysts tested with 50 different substrates under 5 different conditions, with each profile made of 10 points. This matrix contains 50 × 50 ×5 × 10 = 125,000 data points. It merits some serious thinking about data mining, because it is unlikely that you will see anything useful just by looking at 125,000 numbers. PCA can reduce this large matrix into two smaller matrices that are easier to examine and interpret. Using PCA, you can extract the key factors. These are the principal components, or PCs (sometimes also called the latent variables). Each PC is a linear combination of the original variables. But unlike the original variables, which may be correlated with each other, the PCs are orthogonal (i.e., uncorrelated, independent of one another) [10].

Mathematically speaking, if $\mathbf{X}$ is an $(I \times J)$ matrix that contains J variables for I reactions, PCA divides this matrix into a systematic part $\mathbf{TP}^\mathrm{T}$ (the PCA model), and a residuals part $\mathbf{E}$ Eq. (2). $\mathbf{T}$ $(I \times R)$, and $\mathbf{P}$ $(J \times R)$, are two smaller matrices, the size of which depends on R, the number of significant PCs. $\mathbf{T}$ is the scores matrix. It represents the spread of the reactions within the model space. $\mathbf{P}$ is the loadings matrix. It describes the relationships between the variables.

$$\mathbf{X} = \mathbf{t}_1\mathbf{p}_1^\mathrm{T} + \mathbf{t}_2\mathbf{p}_2^\mathrm{T} + \cdots + \mathbf{t}_n\mathbf{p}_n^\mathrm{T} + \mathbf{E} = \mathbf{TP}^\mathrm{T} + \mathbf{E} \tag{2}$$

What PCA actually does is project the data matrix $\mathbf{X}$ onto a lower dimensional space. You start with J variables, and end with R orthogonal PCs (where $R \ll J$). This projection gives a simplified view of the data, highlighting the important variables. Fig. 2 shows an example where two points are projected from a three-dimensional space onto a two-dimensional surface.

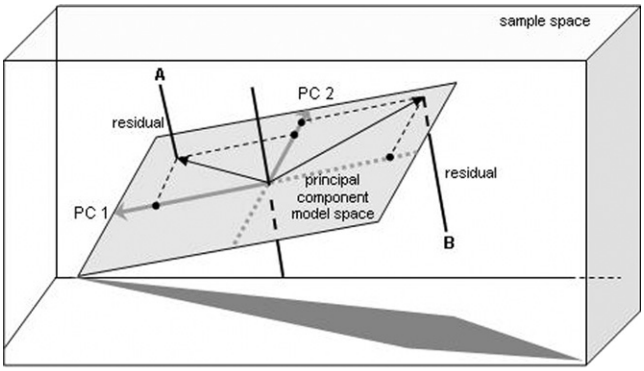

Fig. 2. PCA reduces the dimensionality of the problem by projecting the original dataset onto a lower dimension PC model, in which the new variables are orthogonal to each other. The distance from point **A** to the PCA model space equals the residual value for catalyst **A**.

PCs are ranked according to the fraction of variance of the dataset that they explain. The first PC is the most important (explains the largest fraction of variance), and so forth. Selecting the correct number of PCs is crucial. Too few PCs will leave important information out of the model, but too many PCs will include noise, and decrease the model's robustness (if $R \sim J$, performing a PCA is pointless) [11]. Each time you make a new PCA model, you should examine the residuals matrix **E**. If the residuals are structured, it means that some information is left out. You can also decide on the correct number of PCs by performing a cross-validation (see below), or by examining the percentage of the variance explained by the model.

Note that PCA is scale-dependent, so if the original variables differ by orders of magnitude (e.g. one variable is temperature, in the range 300–500 K, and the other is pressure, in the range 0.2–0.4 bar), the numerically large variables will dominate the first few PCs. You can remove this effect in the pre-processing stage, by scaling all variables to unit variance (also known as autoscaling, see Fig. 3). Starting from your original data matrix $\mathbf{X}$ ($I \times J$), you subtract from each value the column average, and divide the result by the column standard deviation Eq. (3). In this way, the range of each variable is scaled so that ±1 corresponds to one standard deviation.

$$a_{ij} = \frac{x_{ij} - \overline{x_j}}{s_j} \tag{3}$$

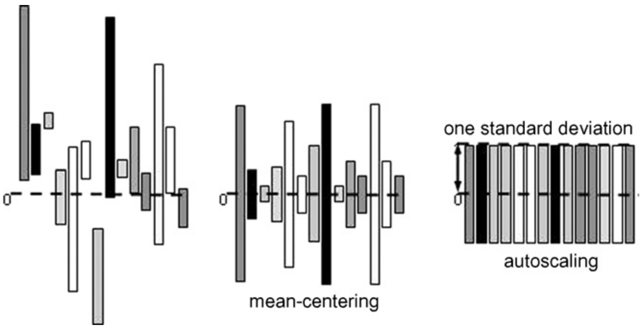

Fig. 3. Schematic representation of mean-centring and autoscaling.

CAUTION! Do not use autoscaling if the variables are all of the same type, because in this case variables with small values may simply describe noise. Autoscaling will inflate these variables, giving them the same weight as the important variables in your dataset.

2.2. Partial least-squares (PLS) regression analysis

PLS regression is a common method for linear modelling of multivariate data [12]. It works with two matrices. The matrix **X** contains the descriptors (the independent variables), and the matrix **Y** contains the figures of merit (the dependent variables). The model approximates the relationship between **X** and **Y** according to Eq. (4), where $\mathbf{B}_{\mathrm{PLS}}$ is the matrix of regression coefficients, and **E** is the matrix of residuals. The regression coefficients can be used for assessing which *x*-variables mainly model a certain *y*-variable. However, with correlated variables, these coefficients are not mathematically independent. To solve this problem, PLS estimates the correlation structure between **X** and **Y** in terms of projections onto a few latent variables, as in PCA. The resulting PLS *x*-weight vectors are used to combine the *x*-variables with the scores, **t**, that predict the *y*-variables. The performance of a PLS model is measured using the explained *y*-variation, R_y^2, and the predicted *y*-variation, q^2 (see also the discussion on model validation in section 4). In general, PLS models are more robust than multiple linear regression and principal component regression methods.

$$\mathbf{Y} = \mathbf{B}_{\mathrm{PLS}}\mathbf{X} + \mathbf{E} \tag{4}$$

The relative importance of each predictor variable in the PLS regression model is calculated using the variable importance parameter (VIP) [13], given by Eq. (5). In this equation, b_{ak} is the regression weight for variable *k* and factor *a*. SSQ_a is the percentage variance captured by latent variable *a*, *n* is the total number of variables and l_v is the number of latent variables used in the regression model. Since the VIP magnitude depends on the number of latent variables, the absolute VIP values are less meaningful than the relative values in a given dataset. Fig. 4 shows an example of a VIP plot for a PLS regression model for 42 bidentate phosphine and phosphite ligands in the hydrocyanation of pentenenitrile, a key step in the production of Nylon 6.6 precursor hexamethylenediamine [13].

$$\mathrm{VIP}_k = \frac{\sum_{a=1}^{l_v} b_{ak}^2 \mathrm{SSQ}_a}{n\sum_{a=1}^{l_v} \mathrm{SSQ}_a} \tag{5}$$

2.3. Artificial neural networks (ANNs)

ANNs mimic the fault-tolerance and the learning capacity of biological neural systems, by simulating the low-level structure of the brain. They are applicable in every situation where there is a relationship between the independent variables (inputs) and predicted variables (outputs), but especially when this relationship is complex and difficult to explain in the usual terms of ‘‘correlations’’. Many catalytic cycles show complex non-linear behaviour, and ANNs are excellent for modelling such

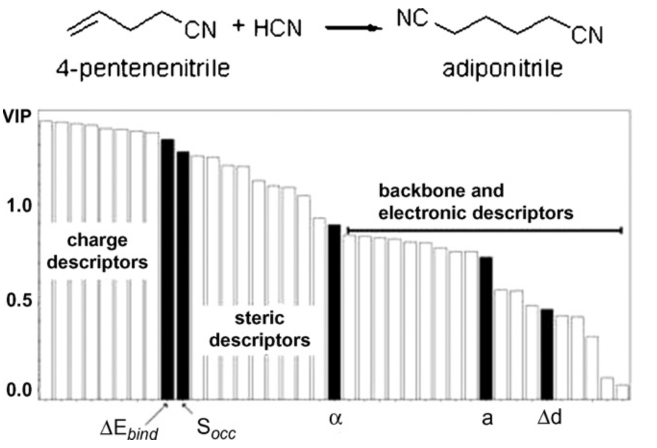

Fig. 4. VIP plot for various descriptors in a PLS model for the hydrocyanation of pentenenitrile in the presence of Ni-biphosphine/biphosphite complexes [13]. Charge descriptors refer to the Mulliken charge calculated at the ligating atoms. ΔE_{bind} is the energy difference between the free ligand and the metal complex, and can be related to the chelating effect and flexibility of the molecule. S_{occ} is the sphere occupation descriptor and measures the sterics around the metal centre. α is the bite angle. a is the second derivative of the flexibility profile polynomial. Δd is the difference in the interatomic distance between the ligating atoms between the free ligand and the complex.

data [14–17]. The disadvantage is that the network is opaque, a ‘‘black box’’—it may give good results, but we cannot follow the ‘‘reasoning’’ behind the model. Furthermore, ANNs perform well when they are well trained, meaning that your training set must be representative of your test set. Another disadvantage of ANNs is that they tend to over-fit, modelling noise as well as real data. This last problem can be avoided with proper model validation (see section 4).

To capture the essence of biological neural systems, an artificial neuron receives a number of inputs, either from original data or from the output of other neurons in the network (Fig. 5). This can be done in different ways, reflecting different network topologies or architectures. There is no golden rule that tells you which topology is most suited for a particular problem, so the best thing to do is simply to try a few different options and compare the results. Eq. (6) shows a general definition, where x is a neuron with n input dendrites x_i and one output axon $y(x)$, and w_i are the weights of these inputs. G is an activation function (usually a sigmoid function or a hyperbolic tangent), based on the sum of the n inputs that determines each neuron’s output threshold.

$$y(x) = G\left(\sum_{i=0}^{n} w_i x_i\right) \tag{6}$$

ANNs have two different phases: a training phase and an execution phase. In the training phase, the weights w_i are determined, and the network is trained to return a specific output when given a specific input. In the execution phase the ANN is fed with new cases (e.g. descriptor values for new catalysts) and returns an output (e.g. a prediction of the figure of merit) on the basis of the network generated in the previous phase. When ANNs are used for regression analysis, confidence levels determine how the network will predict a quantitative value of performance (output value) for each input case. The network’s efficiency is evaluated using the standard deviation ratio, which is the ratio of the prediction error standard deviation to the original output data standard deviation. The lower the ratio, the better the prediction. Note that ANNs can be applied in both regression and classification analysis (see below).

2.4. Classification trees

In classification analysis the variable selection procedure is similar to that of regression, but here the reactions are categorized into ‘positive’ and ‘negative’ cases according to their figure of merit values. This is done by selecting a threshold, e.g. ‘positive cases’ can be defined as those with TON > 1000 and TOF > 500). In each stage, the model splits the data in two. Thus, the first node (also called the parental node) is divided into two child nodes according to the most relevant splitting condition, and so forth. The advantage of

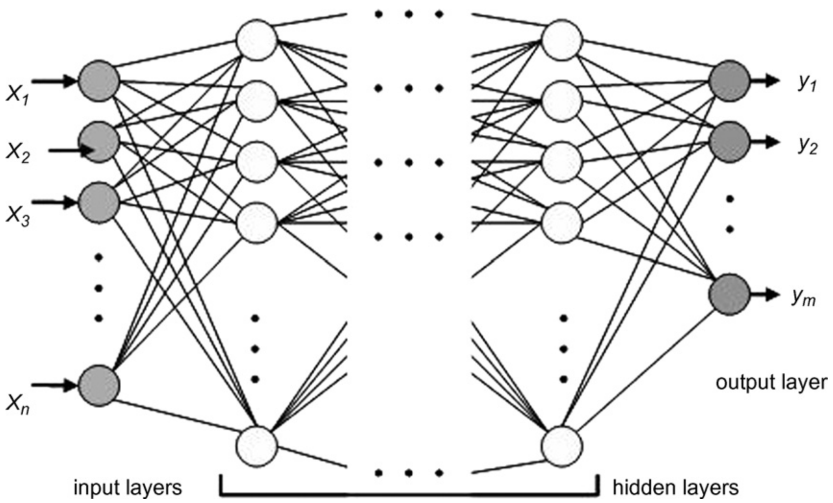

Fig. 5. Schematic drawing of an artificial neural network with a multilayer perceptron topology, showing the pathways from the input x_i to the output y_i, and the ‘‘visible’’ and ‘‘hidden’’ node layers.

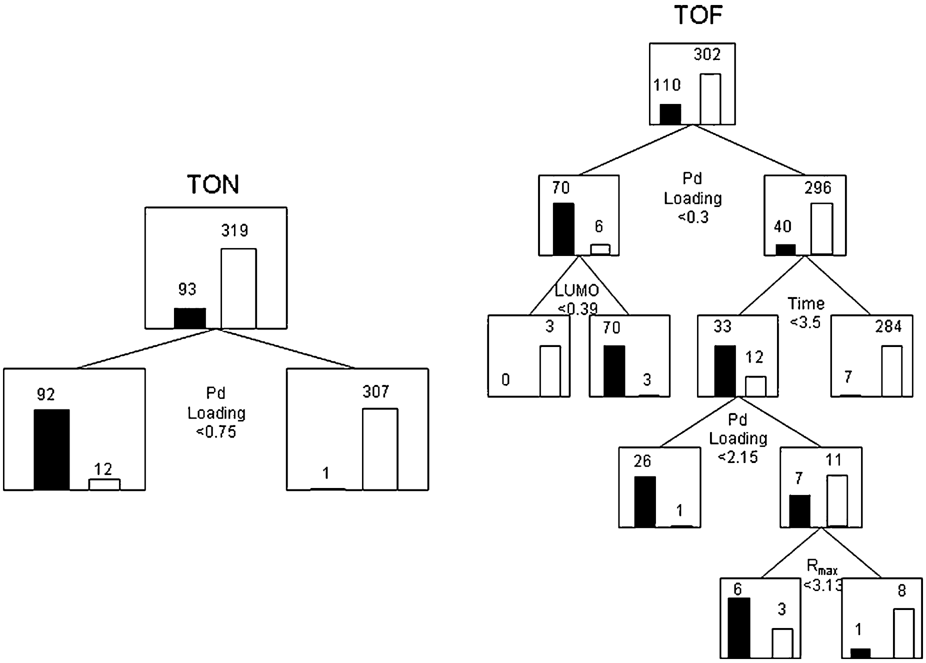

Fig. 6. Classification tree structures of TON (left) and TOF (right), for a dataset of 412 Pd-catalysed Heck reactions described by a total of 74 descriptors [18]. The black and white bars represent positive and negative experiments, respectively. In the case of TON, the most relevant splitting condition is the Pd loading, <0.75%. For the TOF, the first splitting condition is the reaction time, followed by the Pd loading, and the ligand's LUMO energy and R_{max} (the distance between the bulk of the ligand and the metal centre).

classification trees is that, unlike neural networks, one can follow the reasoning behind the model. Fig. 6 shows an example of the TON and TOF classification tree structures for a dataset of 412 Pd-catalysed Heck reactions described by a total of 74 descriptors [18].

3. Data mining and predictive modelling

In silico catalyst optimisation is the ultimate application of computers in catalysis. Imagine a computer program that is fed with data for a given reaction, and outputs the structure of the optimal catalyst for this reaction. Although this fantastic program does not exist (yet), much progress has been made in this direction in the last decade, especially thanks to advances in laboratory automation (for model validation) and data analysis methods. Just like any other new idea, *in silico* catalyst optimisation is accepted by some researchers and met with scepticism by others. Nevertheless, it is essential for realising the potential of high-throughput screening and combinatorial chemistry in catalysis research [19,20]. Computers will not replace chemists, and data mining methods will not replace mechanistic studies. These methods will simply be part of the chemist's toolbox in the 21st century.

3.1. Catalysts, descriptors, and figures of merit

Before making any predictions, we must first understand the problem of catalyst optimisation, and especially define the space in which we want to optimise our data. To do this, we will divide the system in three multi-dimensional spaces, **A**, **B**, and **C** (Fig. 7). Space **A** is a grid containing all the catalyst structures (e.g., if the catalyst in question is a bidentate transition-metal complex, space **A** will contain all of the combinations of transition metal atoms and bidentate ligands,

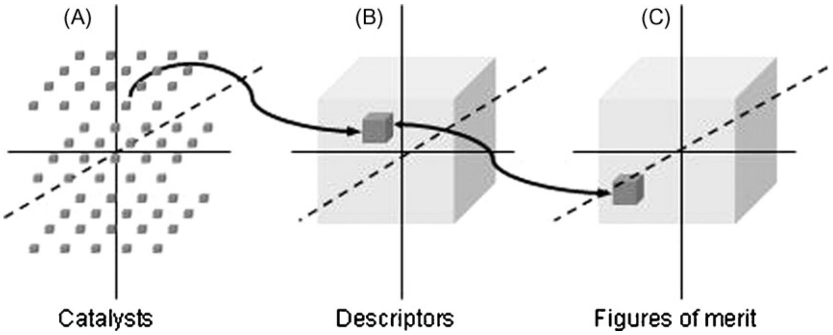

Fig. 7. Three-dimensional simplified representation of the multi-dimensional spaces **A**, **B**, and **C**, containing the catalysts, the molecular descriptor values, and the figures of merit, respectively.

with each point representing a different catalyst); space **B** contains the values of the catalyst descriptors and the reaction conditions (temperature, pressure, solvent type, backbone flexibility, cone angle, partial charge on the metal atom, lipophilicity and so on); and space **C** contains the catalysts' figures of merit (i.e., the TON, TOF, *ee*, price and so forth). In this way, we translated our abstract problem in catalysis to the (still abstract) problem of relating one multi-dimensional space to another. The advantage is that the relationship between spaces **B** and **C** can be quantified using quantitative structure-activity relationship (QSAR) and quantitative structure–property relationship (QSPR) models [21,22]. The key to the model lies in space **B**, the descriptors space. Choosing the right descriptors is critical for the model's success.

3.2. Predictive modelling in homogeneous catalysis

Finding good descriptors is relatively easy in homogeneous catalysis, where the catalyst is usually well defined [19]. Descriptors can be calculated at several levels. 3D descriptors, which are based on optimised geometries, can be calculated using molecular mechanics forcefields and quantum mechanics calculations. Although 3D descriptors offer a realistic representation of chemical systems, their computational cost depends on the system's size and number of degrees of freedom. If you plan to optimise large numbers of catalysts (e.g. virtual libraries for combinatorial optimisation studies), 3D descriptors are simply too costly. In such cases, the simpler 2D descriptors (also called topological descriptors) provide a viable alternative. Topological descriptors are derived directly from molecular connectivity tables, without using any 3D atom coordinates. They are calculated using graph theory, which describes the atom connectivity in hydrogen-suppressed molecules [23,24].

Topological descriptors give information on the molecular size, flexibility, electron distribution and various other physicochemical properties. They are, as Fig. 8 shows, 3–5 orders of magnitude faster than 3D descriptors, depending on the geometry optimisation method used for the latter [25]. Unfortunately, this lower cost is offset by several limitations: First, although 2D descriptors account for specific physicochemical properties, they have no direct heuristic interpretation, because they are far from our 'chemical intuition'. Second, 2D descriptors neglect conformational information, and because they are two-dimensional they cannot be used for modelling chirality.

3.3. Predictive modelling in heterogeneous catalysis

Finding good descriptors for heterogeneous catalysts is much more difficult. Unlike molecular catalysts and organometallic complexes, the activity of solid catalysts depends on a multitude of parameters: different types of active sites, synthesis conditions, thermal treatments, and ageing. Moreover, the properties of many solids can change discontinuously. New phases can form at different compositions, temperatures and pressures, and even the catalyst size can influence the reaction. Nano-sized gold particles, for example, are very different catalysts from 'bulk gold', and two supported gold catalysts can have different activities even when they contain identical amounts of gold and support, respectively [26]. Solid surfaces are anything but uniform, and solid catalysts have a variety of sites. To complicate things further, sometimes the real active sites are not those observed in characterisation studies, but rather metastable defects that are difficult to characterise.

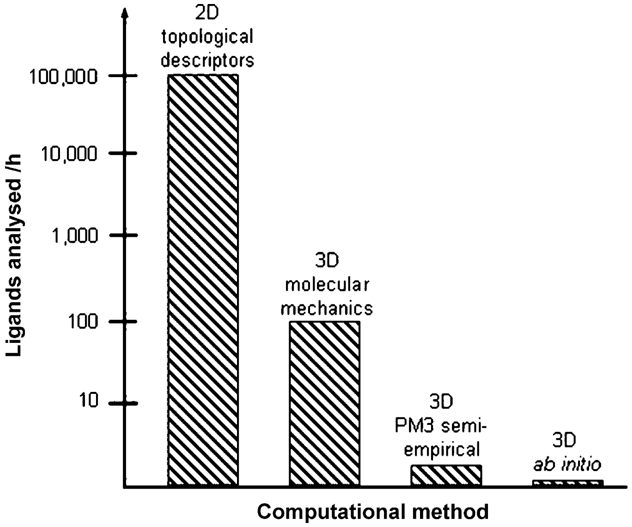

Fig. 8. Bar graph comparing the analysis capacity, in number of ligands'' bite angles and flexibility ranges calculated per hour, using topological descriptors (2D), and 3D descriptors based on MM forcefields, semi-empirical QM calculations, and ab initio QM calculations. All calculations were performed on a 2.5 GHz desktop computer (analysis capacity will improve with better computers and software, but the 2D:3D cost-effectiveness ratio will remain).

Because of this, relying solely on catalyst composition parameters is impractical (compositional descriptors are mainly applicable in cases where the catalyst is a crystalline material, and where the changes in composition do not lead to phase changes [27]). Instead, we need a descriptor toolbox that can account for the discontinuities and nonlinear dependencies. A few promising starts have been made in this direction [28–31]. Klanner et al. combined compositional descriptors and tabulated physico-chemical data, collecting a total of over 3000 descriptors for 467 catalysts, which were then tested in a high-throughput reactor in propene autoxidation. Naturally, such a dataset is over-determined (i.e. there are many more descriptors than data points, so that one can always find good correlations, but these correlations are often meaningless). To obtain meaningful correlations, the researchers needed to discard some descriptors. Interestingly, feature selection algorithms could not select the relevant descriptors, and so a selection was made based on chemical intuition. This is a good example of how chemical knowledge can be integrated in data mining models (see also the following section on data mining in catalysis). A subset of 75 'chemically relevant' descriptors was selected, and prediction models were constructed using artificial neural networks (ANNs) and classification trees (see section 2 for an overview of these methods) [32]. Significantly, both methods could predict "good" or "bad" propene oxidation catalysts. The prediction rate of the ANNs was typically 0.5–0.7, much higher than that of random models (typically 0.2–0.3).

In another study, Corma et al. combined genetic algorithms (GAs) and ANNs for predicting the performance of virtual catalyst libraries for oxidative dehydrogenation of ethane, using the catalyst composition as input parameters [33,34]. The virtual screening was again combined with high-throughput experimentation, using the predictions of the ANNs as a theoretical pre-screening, and thus avoiding the testing of poorly-performing materials. Although the subsets tested were very small compared to the catalyst space, a significant improvement was obtained after 7 generations, demonstrating the potential of this approach. The same group also used high-throughput X-ray diffraction studies to build descriptor datasets for epoxidation catalysts, combining the spectral data with catalyst composition data and creating an automated synthesis, testing, and modelling workflow [35].

3.4. Predictive modelling in biocatalysis

Although enzymes are far too complex for a detailed descriptor modelling at the molecular level, predictive modelling plays an increasingly important role in the search for new biocatalysts [36]. The experimental techniques for designing enzymes that can operate under harsh process conditions (high temperatures, acidic/basic pH, and/or organic solvents) rely heavily on genetic engineering and combinatorial chemistry. These efforts are complemented by a variety of computational screening tools [37]. The main composition variable here is the primary structure (i.e., the amino acid sequence of the protein). Computer algorithms screen the sequence space, eliminating those sequences that are incompatible with the protein folding model, and thus reducing the number of garbage experiments [38]. Protein modelling algorithms can even insert potentially active catalytic residues into ‘‘virtual proteins’’, and search for candidates with an improved binding affinity to high-energy reaction intermediates [39,40]. Examples include the re-designing of active sites for improved catalytic activity [41], and the designing of thermostable enzymes [42].

4. Model validation: separating knowledge from garbage

One problem with computer models, or indeed with the scientists using them, is that when the models do not crash, the scientists tend to believe the results. Without proper validation, however, deducing anything from any model is hazardous business. The model may be over-fitting (finding trends in noise), or predictions may be out of range (extrapolation), leading to ridiculous results. Model validation is like a control experiment in the laboratory. It may be tedious and time-consuming, but it is essential. This section gives the basics of four useful validation approaches. Readers wishing to explore this subject further should consult the work of Tropsha et al. [43].

Regardless of which model you use, the validation procedure depends on the amount and quality of the data. When there is enough data, the set should be divided into three parts: a training set, a test set and a validation set. The model is then constructed using the training set, tested with the test set, and (possibly) improved and re-tested. When you are happy about the model, you can test its performance on the validation set. After this, you cannot tinker with the model again (because it has already ‘‘seen’’ the validation set). If you have insufficient data for three subsets, you divide the data in two: a training set and a test set. You construct the model using the training set, validate it using cross-validation or bootstrapping (see below), and then test its performance with the test set.

4.1. Cross-validation and bootstrapping

Let us assume that we have measured the TON and TOF of 200 catalytic reactions, and calculated a QSAR/QSPR regression model, that connects the catalyst descriptors to the figures of merit. To validate this model, we divide our dataset in two parts: a training set (also known as the calibration set), used for developing the model, and a test set (also known as the prediction set) [12]. We know the figures of merit for the test set, but we do not use them when generating the model. Instead, we calculate the regression equation for the training set (say for $n = 150$ reactions), and using this equation, predict the TON and the TOF for the test set (the remaining 50 reactions). In this way we can compare the performance of different models, all trained on with the same training set. This is known as cross-validation. The model's predictive performance is measured by the cross-validation correlation coefficient, q^2 (Eq. (7), where y_i, $\hat{y}_i$, and $\bar{y}_i$ are the measured, predicted, and average y-values, respectively). Note, however, that although a high q^2 value is a necessary condition for good predictions, it is not a sufficient condition [44]. Cross-validation should be complemented by other methods to ensure the model's robustness and prediction accuracy.

$$q^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \qquad (7)$$

The choice of training and test sets may bias the results [45]. To avoid such problems, we can partition the original set in several different ways (i.e., choose different combinations of training sets and test sets) and compute an average score over the different partitions (leave-n-out cross-validation). An extreme variant of this is splitting the 200 reactions into a training set of size 199 and a test of size 1. This is called leave-one-out cross-validation. The advantage here is that all the data is used for training, without holding any back in a separate test set.

Re-sampling of the dataset in different ways, or bootstrapping, is similar to cross-validation [46]. The idea behind bootstrapping is that the dataset should be representative of the total population. Catalysts from the dataset are selected at random, and divided in subsets. Some catalysts may appear in several subsets, while others may not appear at all. Subsequently, some of the subsets are used for building a model. The results for the remaining catalysts are then predicted using this model. A high average q^2 value indicates the model's robustness.

4.2. Mixing the dependent variables (y-randomising)

Randomising of the y-variables (also known as a permutation test) is a common method for testing the model's robustness. Here, the vector containing the figures of merit is shuffled randomly, so that the figures of merit no longer match to the original descriptor values. A new model is then generated using the original descriptor values, and the process is repeated several times. In principle, this should give models with very low R^2 and q^2 values. Models that fail this negative test should be discarded, because any random collection of values for the figures of merit would do just as well.

Another simple variant of this method is testing the model on completely random data. Generate a random series of numbers for your figure of merit **y**, and then run your model. You should get only noise—if you get meaningful results (i.e., high R^2 and q^2 values) out of random data, then there is something seriously wrong with your model.

4.3. Defining the model domain

Every model has limitations. Even the most robust and best-validated regression model will not predict the outcome for *all* catalysts. Therefore, you must define the application domain of the model. Usually, interpolation within the model space will yield acceptable results. Extrapolation is more dangerous, and should be done only in cases where the new catalysts or reaction conditions are sufficiently close to the model. There are several statistical parameters for measuring this ‘‘closeness’’, such as the distance to the nearest neighbour within the model space (see the discussion on catalyst diversity in [11]). Another approach uses the effective prediction domain (EPD), which defines the prediction boundaries of regression models with correlated variables [47].

5. Summary and outlook

Robotic systems can now test hundreds of catalysts per day, yielding mind-boggling amounts of results. Nevertheless, the total catalyst space is much too large for exhaustive screening, even using robots. Thus, we must choose which areas to search in. Moreover, although high-throughput experimentation gives a lot of data, much of that is ‘garbage data’, that must be sifted out. Statistical methods such as principal component analysis (PCA), partial least squares (PLS), and artificial neural networks (ANNs), can highlight trends in large datasets. Data mining can even indicate regions in the ‘catalyst space’ where ‘good catalysts’’ are likely to be found. This type of predictive modelling, coupled to the power of combinatorial synthesis and screening, puts us on the brink of true *in silico* catalyst design [48].

References

[1] G. Rothenberg, Catalysis: Concepts and Green Applications, Wiley-VCH, Weinheim, 2008, , ISBN: 978-3-527-31824-7.

[2] J.A. Westerhuis, H.F.M. Boelens, D. Iron, G. Rothenberg, Anal. Chem. 76 (2004) 3171–3178.

[3] L.A. Baumes, J. Comb. Chem. 8 (2006) 304–314.

[4] J.M. Caruthers, J.A. Lauterbach, K.T. Thomson, V. Venkatasubramanian, C.M. Snively, A. Bhan, S. Katare, G. Oskarsdottir, J. Catal. 216 (2003) 98–109.

[5] D.L. Massart, B.G.M. Vandeginste, L.M.C. Buydens, S. de Jong, P.J. Lewi, J. Smeyers-Verbeke, Handbook of Chemometrics and Qualimetrics, Elsevier, 1998, , ISBN: 0444828540.

[6] R.L. Tranter, Design and Analysis In Chemical Research, CRC Press, Sheffield, 2000, , ISBN: 1-85075-994-4.

[7] A.L. Fernandez, C. Reyes, A. Prock, W.P. Giering, J. Chem. Soc., Perkin Trans. 2 (2000) 1033–1041.

[8] A. Bocker, G. Schneider, A. Teekentrup, QSAR Comb. Sci. 23 (2004) 207–213.

[9] W.P. Walters, B.B. Goldman, Curr. Opin. Drug Discov. Develop. 8 (2005) 329–333.

[10] S. Wold, K. Esbensen, P. Geladi, Chemom. Intell. Lab. Syst. 2 (1987) 37–52.

[11] J.A. Westerhuis, J.A. Hageman, H.W. Frühauf, G. Rothenberg, Chim. Oggi—Chem. Today 25 (2007) 28–31.

[12] P. Geladi, B.R. Kowalski, Anal. Chim. Acta 185 (1986) 1–17.

[13] E. Burello, P. Marion, J.-C. Galland, A. Chamard, G. Rothenberg, Adv. Synth. Catal. 347 (2005) 803–810.

[14] J.M. Serra, A. Corma, A. Chica, E. Argente, V. Botti, Catal. Today 81 (2003) 393–403.

[15] T.R. Cundari, J. Deng, H.F. Pop, C. Sarbu, J. Chem. Inf. Comp. Sci. 40 (2000) 1052–1061.

[16] T.R. Cundari, M. Russo, J. Chem. Inf. Comp. Sci. 41 (2001) 281–287.

[17] T.R. Cundari, C. Sarbu, H.F. Pop, J. Chem. Inf. Comp. Sci. 42 (2002) 1363–1369.

[18] E. Burello, D. Farrusseng, G. Rothenberg, Adv. Synth. Catal. 346 (2004) 1844–1853.

[19] E. Burello, G. Rothenberg, Int. J. Mol. Sci. 7 (2006) 375–404.

[20] G. Rothenberg, H.F.M. Boelens, D. Iron, J.A. Westerhuis, Chim. Oggi 21 (2003) 80–83.

[21] H. Bönnemann, Angew. Chem. Int. Ed. Engl. 24 (1985) 248–262.

[22] K.D. Cooney, T.R. Cundari, N.W. Hoffman, K.A. Pittard, M.D. Temple, Y. Zhao, J. Am. Chem. Soc. 125 (2003) 4318–4324.

[23] R. Diestel, Graph Theory, vol. 173, Springer Verlag, New York, 2000.

[24] P.D. Iedema, H.C.J. Hoefsloot, Macromol. Theor. Simul. 10 (2001) 855–869.

[25] E. Burello, G. Rothenberg, Adv. Synth. Catal. 347 (2005) 1969–1977.

[26] M. Haruta, N. Yamada, T. Kobayashi, S. Iijima, J. Catal. 115 (1989) 301–309.

[27] J. Beckers, L.M. van der Zande, G. Rothenberg, ChemPhysChem 7 (2006) 747–755.

[28] C. Klanner, D. Farrusseng, L. Baumes, C. Mirodatos, F. Schueth, QSAR Comb. Sci. 22 (2003) 729–736.

[29] D. Farrusseng, C. Klanner, L. Baumes, M. Lengliz, C. Mirodatos, F. Schueth, QSAR Comb. Sci. 24 (2005) 78–93.

[30] M. Holena, M. Baerns, Catal. Today 81 (2003) 485–494.

[31] L. Baumes, D. Farrusseng, M. Lengliz, C. Mirodatos, QSAR Comb. Sci. 23 (2004) 767–778.

[32] C. Klanner, D. Farrusseng, L. Baumes, M. Lengliz, C. Mirodatos, F. Schueth, Angew. Chem. Int. Ed. 43 (2004) 5347–5349.

[33] A. Corma, J.M. Serra, E. Argente, V. Botti, S. Valero, ChemPhysChem 3 (2002) 939–945.

[34] J.M. Serra, A. Corma, S. Valero, E. Argente, V. Botti, QSAR Comb. Sci. 26 (2007) 11–26.

[35] A. Corma, J.M. Serra, P. Serna, M. Moliner, J. Catal. 232 (2005) 335–341.

[36] F.H. Arnold, Nature 409 (2001) 253–257.

[37] G. Hibbert Edward, A. Dalby Paul, Microb. Cell Fact. 4 (2005) 29.

[38] R.J. Hayes, J. Bentzien, M.L. Ary, M.Y. Hwang, J.M. Jacinto, J. Vielmetter, A. Kundu, B.I. Dahiyat, Proc. Natl. Acad. Sci. U.S.A. 99 (2002) 15926–15931.

[39] D.N. Bolon, S.L. Mayo, Proc. Natl. Acad. Sci. U.S.A. 98 (2001) 14274–14279.

[40] M.A. Dwyer, L.L. Looger, H.W. Hellinga, Science 304 (2004) 1967–1971.

[41] J.K. Lassila, J.R. Keeffe, P. Oelschlaeger, S.L. Mayo, PEDS 18 (2005) 161–163.

[42] A. Korkegian, M.E. Black, D. Baker, B.L. Stoddard, Science 308 (2005) 857–860.

[43] A. Tropsha, P. Gramatica, V.K. Gombar, QSAR Comb. Sci. 22 (2003) 69–77.

[44] A. Golbraikh, A. Tropsha, J. Molec. Graph. Model. 20 (2002) 269–276.

[45] A. Golbraikh, M. Shen, Z. Xiao, Y.-D. Xiao, K.-H. Lee, A. Tropsha, J. Comput. Aided Molec. Des. 17 (2003) 241–253.

[46] R. Wehrens, H. Putter, L.M.C. Buydens, Chemom. Intell. Lab. Syst. 54 (2000) 35–52.

[47] J. Mandel, R.W. Gerlach, J. Res. Nat. Inst. Stand. Technol. 90 (1986) 465–478.

[48] J.A. Hageman, J.A. Westerhuis, H.W. Frühauf, G. Rothenberg, Adv. Synth. Catal. 348 (2006) 361–369.