Texture based image classification for nanoparticle surface characterisation and machine learning

Baichuan Sun and Amanda S Barnard

Data61, CSIRO, Door 34 Goods Shed, Village St, Docklands VIC 3008 Australia
1 Author to whom any correspondence should be addressed.
E-mail: baichuan.sun@data61.csiro.au and amanda.barnard@data61.csiro.au

Keywords: nanoparticle, nanomaterial, data reduction, materials informatics, image processing, machine learning

Abstract

Restricting materials informatics to the numerical parameters output from conventional materials modelling software restricts us to a subset of machine learning methods capable of uncovering structure/property relationships and driving materials discovery and design. Presented here is a simple way of converting materials structures into unique image-based fingerprints suitable for image processing methods, that does not require subjective pre-assessment of the data and selection of descriptors by the user. This combination of methods is shown to classify the morphologies in a set of 425 silver nanoparticles in a meaningful way, and predict the correlation with the energy of the Fermi level in agreement with other machine learning methods that required user intervention. Moving to an image-based, rather than feature list-based, description of nanoparticles and materials brings us one step closer to using experimental micrographs as inputs for machine learning.

In recent years materials informatics has become synonymous with computational materials science; not because of the common use of computers, but because much of the data that has been deemed suitable for data-driven analysis has been generated using conventional molecule modelling methods. Computational packages already allow users to compile lists of descriptors that are scientifically intuitive, based on the initial input parameters of the simulations, simple numerical and statistical characterisation of the output structures, or the target properties being considered. However, there is little evidence to suggest that the numerical features conveniently delivered by available software packages are the best inputs for machine learning. While they have been shown to work [1–6], there is also evidence to suggest that features selected by scientist are not always optimal [7, 8] and purely numerical methods can describe materials in ways that are more suitable for machine learning [9, 10]. By restricting ourselves to the traditional characterisation parameters used in computational chemistry, physics and materials science we are also limiting ourselves to a subset of the machine learning methods offer by computer science, and missing opportunities to model experimental data.

To explore new ways of representing materials data that eliminates the need for restrictive lists of user-defined materials features we have used some dimension reduction methods and generated images of the surface texture of a set of nanoparticles to be used as fingerprint descriptors. Mapping high-dimensional data into low-dimensional space aids in visualisation and allows us to see intrinsic patterns in the data, such as structure/property relationships, without needing to classify the structures from the outset. Using image-based (rather than a list of materials-based) descriptors allows us to test established image processing methods, with the future goal of applying them to experimental micrographs. As we will see in the following sections, these models are simple to implement and can recover the important nanoparticle properties without the need for subjective feature selection, and provide a complement to methods currently used in materials informatics.


Methods

t-distributed stochastic neighbour embedding (t-SNE)

To visualise high-dimensional data in a low-dimensional space, a nonlinear dimensionality reduction technique was developed by van der Maaten and Hinton using t-distributed stochastic neighbour embedding (t-SNE) [11]. For \( n \) objects \( x_i \) in \( d \) dimension, the conditional probability that \( x_i \) would be picked as a neighbour given \( x_j \) is expressed as:

\[
P(x_j|x_i) = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2 / 2\sigma_i^2)},
\]

where \( \sigma_i \) is the variance of the Gaussian that is centred on \( x_i \). \( \sigma_i \) is adapted to the density of the data such that smaller values are used for data in high density. The joint probabilities in the high-dimensional space is then set as:

\[
P(x_i, x_j) = \frac{P(x_j|x_i) + P(x_i|x_j)}{2\pi}.
\]

For the low-dimensional counterparts, \( y_i \) of the high-dimensional data \( x_i \), a student t-distribution with a degree of freedom one is used to compute the joint probabilities \( Q(y_j, y_i) \) as,

\[
Q(x_i, x_j) = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq i} (1 + \|y_i - y_k\|^2)^{-1}}.
\]

This distribution is heavy-tailed so that dissimilar objects in low-dimensional space can be modelled even if they are far apart. The optimisation objective is to have \( P(x_j, x_i) \) and \( Q(x_j, x_i) \) be equal, which means to minimise the cost function \( \sum_{i \neq j} P_{ij} \log(P_{ij}/Q_{ij}) \).

Self-organising map (SOM)

A Kohonen network, or SOM [12], is an unsupervised artificial neural network (ANN) [13] mapping of nonlinear multi-dimensional spaces into low-dimensional (e.g. 2D) spaces that retains the intrinsic topological relationship of the input dataset. SOMs are ideal for representing 3D information as a single 2D snapshot, to form a base for further analysis. For instance, SOMs were used by Gasteiger et al [14] to generate 2D topological feature maps of the electrostatic potential (ESP) on molecular surfaces. These were then treated as fingerprints to define the molecular similarity.

The basic unit of a SOM is the neuron, or cell, which can be organised as square or hexagonal grids. The square model lacks rotation invariance when counting more than four neighbours and is therefore regarded as anisotropic; the hexagonal grid model has a symmetry of six neighbour neurons which overcomes such anisotropy. All neurons can be arranged in a planar (approximate) rectangle in both cases. However, the neurons on the edge will have smaller weights as they are far from all other cells in Euclidean distance, which reduces the symmetry of the resultant SOM. To overcome this issue, periodic boundary condition can be applied when calculating the Euclidean distance between neurons, to connect the upper edge with the lower edge, and the right border with the left border. As a result the SOM plane essentially occupies the surface of a torus.

The weights of all neurons are initialised to normalised vectors of the same dimension of the original data. For the 3D mesh surface sampling of the molecular ESP, for example, the dimension for each vector is three. One could initialise these vectors with random numbers, but this does not guarantee consistency once the original data is extrinsically modified, such as during translation or rotation in the 3D space. Different SOMs could be obtained even though these are intrinsically the same physical structure. Therefore the SOM neuron weights should be initialised based on intrinsic attributes of the input data, which is invariant against rotation or translation. By initialising the neuron weights according to a `plane’ constructed by the first two eigenvectors derived from principle components analysis of the data entry, the initial weights are unique for different molecular mesh surfaces, and are consistent no matter how the molecule translates or rotates in the 3D space.

During each training step every neuron competes against all others until each original data point finds the one neuron that is closest to it in Euclidean space. The selected neuron is known as the best matching unit (BMU). Suppose the input data point is \( x \), and the weight of neuron \( i, j \) is \( w_{ij} \), then the Euclidean distance is:

\[
\text{Dist} = \sqrt{\sum_{v=1}^{3} (x_v - w_{ij}^v)^2},
\]

where \( v \) is each component of the 3D vector. Once the BMU is located, all neurons centred on it will be updated with their weights. At the beginning of training all neurons on the SOM will be updated, and at the end of training only the BMU will be adjusted. This means the radius of neighbourhood \( \delta(t) \) decreases with each...
subsequent iteration step, \( t \). Amongst other options, a linear relation can be adopted with a given total epochs \( n_{epo} \) to train the SOM, as:

\[
\delta(t) = \delta_0 - \frac{\delta_n}{n_{epo}} \cdot t
\]

(5)

Here, \( \delta_0 \) is the initial radius, which by convention, can be set to be half size of the SOM. \( t \in 2, 3, 4, ..., n_{epo} \) is determined by the boundary condition such that it equals one at the last step of training. In addition to decay of the neighbourhood radius, the learning rate \( L(t) \) for updating weights on each neuron also decreases with each iteration of \( t \). At the beginning the weights need to be updated faster, while in the last few steps the weights should be modified more slowly. Such a process is expressed below:

\[
L(t) = L_0 - \frac{L_n}{n_{epo}} \cdot t
\]

(6)

where \( L_0 \) is the initial learning rate, e.g. 0.1, and \( L_n \) is the constant which is determined by the boundary condition such that the last iteration of training is very fine grained. As a result, the overall weights updating formula can be given as:

\[
\mathbf{w}(t + 1) = \mathbf{w}(t) + L(t) \exp \left[ -\frac{\mathbf{dist}^2}{2\delta^2(t)} \right] \mathbf{x} - \mathbf{w}(t)
\]

(7)

dist is the Euclidean distance of a neuron at \( i, j \) from the BMU of \( \mathbf{x} \) at step \( t \).

Another hyper-parameter of SOM training is the total number of epochs, \( n_{epo} \). If the number is too small the SOM will be under-trained, with all dissimilar data entries gathering at close regions on the SOM. This means the SOM fails to represent the data set with high resolution. If the number is too large the SOM will be over-trained leading to a waste of computational resources. It is possible to measure the number of void neurons which have no weight, and stop the training while this number stops decrease over a threshold after, for instance, 5 epochs.

**Image texture recognition**

Once a 3D structure is converted to a 2D image a range of robust image processing methods become available. Local binary patterns (LBP) extract feature descriptors from images for texture pattern recognition [15, 16] which can be used to analyse the nanoparticle ESP images. LBP first converts colour images into grayscale, then updates the pixel values by comparing the central pixel value with all the neighbouring pixels. If the current pixel is greater or equal to the neighbouring value, the corresponding bit is set to 1; otherwise to 0. Then the binary pattern, 10110110, is converted to a decimal number 182 to update the central pixel.

![Local binary pattern (LBP) computing process. The central pixel is 7, and it is compared to all its neighbours (8 in this case) in counter-clockwise direction. If the neighbourhood pixel value is greater or equal to 7, its corresponding bit in the binary array is set to 1; otherwise to 0. Then the binary pattern, 10110110, is converted to a decimal number 182 to update the central pixel.](image)

Results and discussion

The computational dataset used in this study contains 425 silver nanoparticles [17] with a diameter between 0.5 and 4.9 nm (13–2947 atoms), and a range of different morphologies defined by zonohedrons enclosed by...
fringes are well resolved

ambiguous, particularly at small sizes, due to the restrictions imposed by the crystallographic lattice. It is also

morphology, one of the aims of this work is to eliminate this step, since the geometry assignment can be

a face centered cubic or twinned face centered cubic shape. Although it is possible to assign a shape to each

each particle has been extensively reported elsewhere [18, 25], and is included in the dataset available for download [17]. Any group of features could be chosen, but the aim here is to test our methods and so a diverse but intuitive group is desirable. The numerical values of all features are standardised by removing the mean and scaling to unit variance. Without proper standardisation, machine learning algorithms (including t-SNE) become biased by features with higher absolute variance, resulting in statistical errors.

By mapping high-dimensional data to 2D using t-SNE, we examine how the structured features listed in
table 2 can describe different geometric shapes (structure) and the energy of the Fermi level (property) [18]. The results are illustrated in figure 2, where we can see that nanoparticles mapped together tend to be of the same or similar shapes (figure 2(a)), and have a similar Fermi energy (figure 2(b)). This is consistent with previously reported results on the impact of shape distributions on the electron charge transfer properties of silver nanoparticles [18], and confirms that there is a shape-dependent structure/property relationship, even though we have not explicitly included the shapes as a descriptor. Specifically, the icosahedral (4), Ino decahedral (17) and Marks decahedral (18) nanoparticles cluster together (see top-left of figure 2) and together exhibit lower Fermi energies. These morphologies are also heavily twinned. The cuboctahedral (1), octahedral (8) and truncated octahedral (14) nanoparticles also exhibit lower Fermi energies, but map together separately (see bottom-right of figure 2). Even though both groups have high [111] surface area and similar Fermi energies, the t-SNE method recognises the different internal (bulk) structures. The remaining shapes lack twinning and are mapped adjacent to the cuboctahedra, octahedra and truncated octahedra (bottom-right of figure 2) but exhibit a higher Fermi energy. The only exceptions are the tetrahedron (12) and simple decahedron (19) which are separated; both shape being characterised by very acute edges and corners. The intuitive list of materials-based features is suitable and sufficient to classify different shapes (structures) and Fermi energies (properties) simultaneously.

| Reference number | Geometric shape                      |
|-------------------|--------------------------------------|
| 1                 | Cuboctahedron                         |
| 2                 | Great rhombicuboctahedron             |
| 3                 | Hexakis octahedron                    |
| 4                 | Icosahedron                           |
| 5                 | Truncated cube                        |
| 6                 | Rhombic dodecahedron                  |
| 7                 | Modified truncated octahedron         |
| 8                 | Octahedron                            |
| 9                 | Truncated rhombic dodecahedron        |
| 10                | Rhombi-truncated octahedron           |
| 11                | Small rhombicuboctahedron             |
| 12                | Tetrahedron                           |
| 13                | Tetraakis hexahedron                  |
| 14                | Truncated octahedron                  |
| 15                | Triaxis octahedron                    |
| 16                | Trapezohedron                         |
| 17                | Ino decahedron                        |
| 18                | Marks decahedron                      |
| 19                | Simple decahedron                     |

{100}, {110}, {111}, {210}, {113}, {331} and {123} [18], as detailed in table 1. Each structure is fundamentally a face centered cubic or twinned face centered cubic shape. Although it is possible to assign a shape to each morphology, one of the aims of this work is to eliminate this step, since the geometry assignment can be ambiguous, particularly at small sizes, due to the restrictions imposed by the crystallographic lattice. It is also intrinsically difficult to assign 3D morphologies to experimental images, even when the outline and lattice fringes are well resolved [19–24]. Therefore, the first step is to demonstrate that we can successful predict both structures and properties using only objective numerical nanoparticle features.

Feature selection

Each individual Ag nanoparticle has been characterised for its structural and morphological features, as listed in table 2 based on a logical set of descriptors common in computational and experimental materials science. We can see in this list that there are eight features capturing the local bonding configurations (examples of chemical information typically obtained computationally), six features capturing the bulk structures (examples of materials information that could be obtained from diffraction) and ten features capturing surface structures (examples of nanoscale information that could be potentially obtained from spectroscopy). The energetics (stability) of each particle has been extensively reported elsewhere [18, 25], and is included in the dataset available for download [17]. Any group of features could be chosen, but the aim here is to test our methods and so a diverse but intuitive group is desirable. The numerical values of all features are standardised by removing the mean and scaling to unit variance. Without proper standardisation, machine learning algorithms (including t-SNE) become biased by features with higher absolute variance, resulting in statistical errors.

| Reference number | Geometric shape |
|-------------------|-----------------|
| 1                 | Cuboctahedron    |
| 2                 | Great rhombicuboctahedron |
| 3                 | Hexakis octahedron |
| 4                 | Icosahedron      |
| 5                 | Truncated cube   |
| 6                 | Rhombic dodecahedron |
| 7                 | Modified truncated octahedron |
| 8                 | Octahedron       |
| 9                 | Truncated rhombic dodecahedron |
| 10                | Rhombi-truncated octahedron |
| 11                | Small rhombicuboctahedron |
| 12                | Tetrahedron      |
| 13                | Tetraakis hexahedron |
| 14                | Truncated octahedron |
| 15                | Triaxis octahedron |
| 16                | Trapezohedron    |
| 17                | Ino decahedron   |
| 18                | Marks decahedron |
| 19                | Simple decahedron |

Table 1. Silver nanoparticle morphologies, and reference numbers.
Surface texture

To further eliminate the need for subjective assessments and feature selection SOMs were trained directly from the 3D coordinates of the mesh around the molecular surface of the ESP to generate 2D topological maps where neighbouring 3D regions activate SOM cells in close proximity. The SOM cells were coloured to reflect the ESP, as illustrated in figure 3, but in principle any kind of surface texture images could be suitable and is worth investigation.

With the SOM images for each Ag nanoparticle texture analysis was conducted using LBP algorithm (radius of 5 and resolution of 40). As highlighted in figure 4 the normalised histogram encodes the proportion of edge, surface and corner features on the SOM images. Given that each SOM is a direct representation of the surface texture, which is determined by the combination of global features such as the overall size and shape, as well as local features such as the atomic configuration of the surface layer and the surface chemistry, it is highly unlikely

Table 2. Silver nanoparticle features, as descriptors for machine learning.

| Abbreviation | Feature                          |
|--------------|----------------------------------|
| NUM          | Number of atoms                  |
| DEN          | Mass density                     |
| AgCN         | Ag–Ag coordinate number          |
| VA           | Volume per atom                  |
| ABM          | Ag–Ag bond mean length           |
| ABV          | Ag–Ag bond length variance       |
| AGM          | Mean value (Angle Ag–Ag–Ag)      |
| AGV          | Angle Ag–Ag–Ag angle variance    |
| AD           | Average diameter                 |
| AR           | Aspect ratio                     |
| FCC          | Face centered cubic population   |
| HCP          | Hexagonal close packed population|
| ICO          | Icosahedral population           |
| DIS          | Disordered population            |
| NSA          | Number of surface atoms          |
| SCN3         | Surface coordination number 3    |
| SCN4         | Surface coordination number 4    |
| SCN5         | Surface coordination number 5    |
| SCN6         | Surface coordination number 6    |
| SCN7         | Surface coordination number 7    |
| SCN8         | Surface coordination number 8    |
| SCN9         | Surface coordination number 9    |
| SCN10        | Surface coordination number 10   |
| SCN11        | Surface coordination number 11   |

Figure 2. t-SNE plot of Ag nanoparticle dataset using structured features: (a) coloured by shape; (b) coloured by the normalised energy of the Fermi level.
that two different particles could produce the same SOM. In our case each nanoparticle in the ensemble has different surface textures, which produce unique histograms and provide a fingerprint (figure 5), but if two SOMs were to be computationally indistinguishable, it would be because, from the exterior, the two original particles were indistinguishable as well. Provided the basis for classifying the particle is the surface properties, this eliminates the human bias inherent in subjective characterisation of nanoparticle surface features, and provides an objective basis for classification and regression purposes. If the basis for classifying the particle is the bulk features and properties then a surface texture map is clearly inappropriate.

The texture fingerprints of the Ag dataset then mapped to 2D space using t-SNE technique, as shown in figure 6, to see if the SOM descriptor is as reliable as the materials-based feature list. Once again similar shapes are mapped close to each other, and correspond to similar Fermi energies due to the intrinsic structure/property relationship. The cuboctahedron (1), icosahedron (4), octahedron (8), truncated octahedron (14), Ino decahedron (17) and Marks decahedron (18) nanoparticles, dominated by \{111\} facets are mapped close together.
(see bottom-right region in figure 6(a)), with only a few exceptions (see top-left region). The materials-based descriptors identified that these shapes also exhibit a low Fermi energy, and we can see from figure 6(b) that the same structure/property relationship is recovered using SOM. Once again, the remaining shapes are mapped together in other regions of the configuration space (top-left region) with higher Fermi energies; and the tetrahedron (12) and simple decahedron (19) are distinct (bottom-right region). Although the distribution of the nanoparticles in this t-SNE plot appears different than in figure 2 the underlying pattern of information with respect to the structures and properties is preserved. This indicates ESP texture fingerprints are also suitable and sufficient to classify structures and properties, without the need for the subjective selection of nanoparticle features.
Classification and regression

To quantify the accuracy of the surface texture fingerprints for predicting Ag nanoparticle properties machine learning models were built and tested. The shapes classified with low Fermi energies were labelled as class 0, including the cuboctahedron (1), icosahedron (4), octahedron (8), truncated octahedron (14), Ino decahedron (17) and Marks decahedron (18). All other shapes were labelled as class 1. The classification model (using XGBoost [26]) was trained and cross-validated on 340 data point and tested on remaining 85 data points. The classification confusion matrix of the testing result is shown in figure 7. As we can see the accuracy and recall using the surface texture fingerprints exceeded 90%. This confirms that surface texture fingerprints is numerically reliable for Ag nanoparticle Fermi energy prediction.

We then treated all of the materials-based features listed in table 2 as target values to be predicted by surface texture fingerprints, to examine how this single descriptor can be used to model different bulk or surface features of Ag nanoparticles. All the regression models were built, tested and selected using genetic programming (TPOT [27, 28]) without human interference, and the result is shown in figure 8. Here we can see that surface texture fingerprints predicts VA with a coefficient of determination $R^2$ of 94% on testing data set, but performs poorly on bulk features including FCC population, HCP population, ICO population. This is not surprising, given that...
they are surface texture fingerprints. In contrast the surface coordination numbers (SCN3 through to SCN11) are predicted relatively well; SCN6, SCN7, SCN9 and SCN10 being the most accurate. It should be noted that SCN11 is constant zero except for 2 data points and therefore the high testing score does not justify a robust correlation. This observation suggests that bulk-like features and surface features should be considered as complements to each other, rather than replacements, for this type of nanoparticle modelling. To more accurately model bulk-like features one would ideally develop bulk-texture maps.

Conclusions

Using established dimension reduction methods we have shown that surface texture images and image processing method can be used to classify nanoparticles, identify structure/property relationships and recover important materials-based features that would traditionally be used as input descriptors. Using the t-distributed stochastic neighbour embedding method we reduced 24 dimensions to 2, and identified classes of nanoparticles shapes characterised by the degree of twinning and the orientation of the surface facets (which was never an input descriptor) with similar Fermi energies. Surface texture fingerprints were generated using SOMs of the surface ESP of 425 silver nanoparticles and used to model the shape-dependent energy of the Fermi level with an F1 score of 91%. We found that surface texture fingerprints correlate well to surface features such as the type of under-coordinated surface atoms, the volume per atom and the degree of particle anisotropy, but should be used with caution when predicting bulk properties, such as different crystal lattice structures. Surface texture fingerprints we predict a strong relationship between the energy of the Fermi level and the fraction of surface atoms with surface coordination numbers of 6 and 9. These findings are consistent with relationships predicted with materials-based descriptors, ensemble filtering [18] and a combination of k-mean, logistic regression, random forests, principle component analysis and a concise 3 layer ANN [25].

One of the advantages of using image-based descriptors generated in this way is the elimination of human subjectivity, but at this point researchers are still required to identify the filters needed to describe the textures. Should sufficient data become available (> 10 000 points) one could consider using deep learning to obtain these filters automatically, or draw out other image features that make filtering the textures unnecessary. While there is more work to be done the ability to use 2D images and established image processing methods brings us one step closer using experimental images from microscopy as descriptors for machine learning.

Acknowledgments

Computational resources for this project have been supplied by the Australian National Computing Infrastructure national facility under Grant q27, and Western Australia Pawsey Supercomputing Centre Athena project pawsey0238. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

ORCID iDs

Amanda S Barnard ★https://orcid.org/0000-0002-4784-2382

References

[1] Ramprasad R, Batra R, Pilania G, Mannodi-Kanakkithodi A and Kim C 2017 Comput. Mater. 3 54
[2] Jain A, Hautier G, Ong S P and Persson K 2016 J. Mater. Res. 31 977
[3] Ramakrishnan R and von Lilienfeld A 2017 Rev. Comput. Chem. 30 225
[4] Fernandez M and Barnard A S 2015 ACS Nano 9 11980
[5] Agrawal A and Choudhary A 2016 API Mater. 4 053208
[6] Ward L and Wolverton C 2017 Curr. Opin. Solid State Mater. Sci. 21 167
[7] Swann E, Fernandez M, Coote M L and Barnard A S 2017 ACS Comb. Sci. 19 544
[8] Swann E, Sun B, Cleland D M and Barnard A S 2018 Mol. Simul. 44 905–920
[9] Fernandez M, Shi H Q and Barnard A S 2016 Carbon 103 142
[10] Fernandez M, Abreu J I, Shi H Q and Barnard A S 2016 ACS Comb. Sci. 18 661
[11] Van Der Maaten L J P and Hinton G E 2008 J. Mach. Learn. Res. 9 2579
[12] Kohonen T 1998 Neurocomputing 21 1
[13] Bishop C 1995 Neural Networks for Pattern Recognition (New York: Oxford University Press)
[14] Gasteiger J, Li X, Rudolph C, Sadowski J and Zupan J 1994 J. Am. Chem. Soc. 116 4608
[15] Wang L and He D-C 1990 Pattern Recognit. 23 905
[16] Ojala T, Pietikäinen M and Harwood D 1996 Pattern Recognit. 29 51
[17] Barnard A S and Sun B 2017 Silver Nanoparticle Data Set CSIRO Data Access Portal 2017
[18] Sun B and Barnard A S 2017 Nanoscale 9 12698
[19] Barnard A S, Lin X M and Curtiss L A 2005 *J. Phys. Chem.* B **109** 24465
[20] Ahrenkiel S P, Yu P R, Murphy J E, Nedeljković J M and Donohoe B S 2008 *J. Microsc.* **230** 382
[21] Barnard A S, Young N, Kirkland A I, van Huis M A and Xu H 2009 *ACS Nano* **3** 1431
[22] Chang L Y, Barnard A S, Gontard L C and Dunin-Borkowski R 2010 *Nano Lett.* **10** 3073
[23] Barnard A S, Konishi H and Xu H 2011 *Catal. Sci. Technol.* **1** 1440
[24] Boyd R D, Gunnarsson R, Pilch I and Helmersson U 2014 *J. Phys.: Conf. Ser.* **522** 012065
[25] Sun B, Fernandez M and Barnard A S 2017 *J. Chem. Inf. Model.* **57** 2413
[26] Chen T and Guestrin C 2016 Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (NY, USA) p 785
[27] Olson B S, Bartley N, Urbanowicz R J and Moore J H 2016 *Proc. Genetic and Evolutionary Computation Conf.* (NY, USA) p 485
[28] Olson R S, Urbanowicz R J, Andrews P C, Lavender N A, Kidd L C and Moore J H 2016 Automating Biomedical Data Science Through Tree-Based Pipeline Optimization *Proc. 19th European Conf., Part 1* ed G Squillero and P Burelli (Berlin: Springer) p 123