In too deep? Neural networks versus linear models for critical micelle concentration prediction

Alexander Moriarty,[*,†] Takeshi Kobayashi,[†] Matteo Salvalaglio,[†] Alberto Striolo,[†,‡] Andrew Campbell,[¶] and Ian McRobbie[§]

[†]*Department of Chemical Engineering, University College London, UK*

[‡]*Gallolgy College of Engineering, University of Oklahoma, USA*

[¶]*Physical Chemist, Innospec Ltd., Ellesmere Port, UK*

[§]*Senior Vice President, Research and Technology, Innospec Ltd., Ellesmere Port, UK*

E-mail: alexander.moriarty.21@ucl.ac.uk

Abstract

TODO: Write this.

Introduction

Perhaps the most well established predictor for critical micelle concentration (CMC), X_{cmc}, is the Stauff-Klevens relationship, first published in 1953.[1] It formalised the observation that CMC decreases exponentially with an increase in the number of carbons in the hydrocarbon tail, n_c:

$$\log X_{cmc} = A - Bn_c, \quad B > 0 \tag{1}$$

where A and B are empirical constants that depend on the temperature and the homologous series, i.e. the headgroup. The model is simple, yet accurate, and it is easily interpretable: to reduce CMC, extend the surfactant's hydrocarbon tail. Its only drawback as a predictive model is its very limited applicability domain; each set of parameters is only applicable to surfactants with a specific headgroup and a linear carbon tail. One of the goals of quantitative structure-property relationship (QSPR) development is to produce models that are general, so that we can apply them to a diverse range of compounds, design novel molecules with target properties, and interpret the models' results to glean chemical insights.

To that end, there have been a wealth of investigations into making more general models for CMC prediction.[2–4] (More references to be added.) Recently, an approach based on graph neural networks (GNNs) has produced highly accurate predictions, whilst having the advantage of being applicable to nonionic, cationic, anionic and zwitterionic surfactants simultaneously.[5] Such neural networks have many trainable parameters and a complex functional form. This ensures their versatility as general approximators, but makes them highly susceptible to overfitting (CITATION). By using such complex models, we also abandon the parsimony exhibited by Stauff-Klevens, and chemical insights can be much more difficult to derive. Furthermore, deep neural networks' ostensible 'universality' can be misleading: extrapolating the model's results to out-of-domain molecules (ones that are 'dissimilar' from the training data) will yield unreliable and potentially misleading predictions.

In this article, we assess a spectrum of empirical models for CMC prediction, with the intent of finding the optimum trade-off between accuracy, generality and interpretability for this task, by correlating our models' interpretations with published research. We also apply a technique for adding uncertainty quantification to GNNs, which can indicate whether a molecule is within the model's applicability domain and therefore whether a given prediction is reliable. We aim to demonstrate a best-practices approach to applying machine learning to QSPR tasks on small datasets.

Method

A dataset of 202 surfactants was used, which was curated by a previous work[5] by accumulating results from several publications. This dataset was selected because, to the authors' knowledge, it is currently the largest public dataset of CMCs for several classes of surfactant collected at standard conditions, in an aqueous environment between 20 °C to 25 °C. These data were split into training and test subsets, to simulate the real-world scenario of using a model to make inferences about molecules for which no data is available. The training data were used to fit the models, whilst test data were 'locked away' until it had been decided that the model was optimised, and the performance metrics on the test data were used for comparison. For some models, the training data were further split into optimisation and validation subsets; the optimisation data were used when calculating the loss function during model fitting and the validation data were used for on-the-fly evaluation of model performance during training.

To provide a consistent benchmark of model performance, the same train/test data splits were used as Qin et al.[5] and models were also trained and evaluated using only the nonionic surfactants from the dataset, so as to test whether generalised all-surfactant models can be as accurate as models trained on one class of surfactant. The number of each class of surfactant in the train and test subsets of the data are shown in Table 1.

Table 1: The number of each class of surfactant contained in the train/test subsets of the CMC dataset.

Data subset		Number of			
Surfactant classes	Train/test	Nonionics	Anionics	Cationics	Zwitterionics
All	Train	110	30	31	9
	Test	12	4	4	2
Nonionics	Train	110			
	Test	12			

A QSPR pipeline requires choosing two essential functions: a representation function, whose parameters are defined before training the model, and a mathematical form that maps this representation to a prediction. The processes by which these functions are developed are called *feature engineering* and *model selection*, respectively.

Feature engineering

The ideal molecular representation depends on the task at hand. Ideally, it should be compact, but complete;[6,7] 'as simple as possible, but not simpler.' To that end, the representation should contain enough information to distinguish between isomers that with distinct properties, though concessions can be made if we restrict the model's domain and self-impose limits on the type of isomers we expose the model to, both during training and in use. Representations may also include descriptions of state, such as temperature and pressure,[8] but this is redundant in cases where the training data spans a very limited range of states.

In the case of the Stauff-Klevens model, the representation effectively has two components: the category of the headgroup, and the length of the tail group. The model technically distinguishes between isomers by imposing strict constraints on the structure of the molecules to which it can be applied: position and functional isomers correspond to distinctive categories, so that the model must learn independent parameters for each of them, and the constraint on the tail group means that chain isomers or the presence of non-alkyl groups in the carbon chain are not permitted. Mathematically, this representation can be formalised using a technique called *one-hot encoding* and by defining the set of headgroups for which we have data, $\{h_i \mid 0 \leq i \leq N\}$. The encoding is a vector given by

$$\vec{s}_i = \begin{cases} 1 & \text{if headgroup is } h_i \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Different headgroups therefore correspond to orthogonal encodings. If we have a set of trained parameters $\{A_i, B_i\}$ corresponding to headgroup h_i, Equation 1 can be rewritten as

$$\log X_{cmc} = \mathbf{W}\vec{s} \cdot \begin{bmatrix} 1 \\ -n_c \end{bmatrix}, \quad \mathbf{W} = \begin{pmatrix} A_1 & A_2 & \dots & A_N \\ B_1 & B_2 & \dots & B_N \end{pmatrix}. \tag{3}$$

We can try to make the approach more general by decomposing a molecule into smaller sets of *atomic environments* and representing it by the number of each of these constituents. Because certain groups of atoms and bonds are common in organic surfactants, the resulting feature vectors are not orthogonal, and we can apply the model even when we have made small changes to the headgroup, or introduce branching and other functional groups to the tail.

Extended-connectivity fingerprints

In this approach, the molecule is split into atomic environments up to a given radius, r: each environment is centred on an atom and extends r steps along connecting bonds. Effectively, we discard the categorical encoding in favour of introducing more continuous, count-based features, like n_c. The set of all environments in the training data up to radius r, $\{e_i \mid 0 \leq i \leq N\}$, is extracted and the resulting feature vector is

$$\vec{c_i} = \text{Count}(e_i). \tag{4}$$

Now, a change in headgroup composition is reflected in a change in subgraph counts, and provided the new subgraph exists in our training data, the model can adjust its prediction accordingly. Branch points in a carbon chain are distinguished from main-chain groups, as they terminate in a CH group, rather than CH_2. This type of representation is called an *extended-connectivity fingerprint* (ECFP).[9]

However, these fingerprints do not necessarily distinguish between all positional isomers or chain isomers, particularly with smaller values of r, nor are stereoisomers treated differently. Another potential disadvantage is that the number of unique atomic environments is potentially very large relative to the size of the data available, which poses a risk of overfitting. Furthermore, larger environments necessarily envelop smaller ones, which means that there is some duplicate information in the representation: the presence of a $(CH_2)_3$ environment implies the presence of three CH_2 environments, so that there is multicollinearity. This redundancy can impede model fitting and interpretation.

Molecular graph representation

Both of the approaches described so far rely on molecular feature vectors that cannot describe the molecule's topology. A molecular graph is a structure that achieves this, and it is a popular choice for cheminformatics as well as visualisation of molecular structure. In this approach, each atom is considered a *node* and each bond an *edge*. Rather than having a single feature vector to describe the molecule as a whole, each atom is assigned its own feature vector based on properties such as its element, hybridisation state, charge, etc. These feature vectors are concatenated into a node feature matrix. The graph's structure is then defined by a binary adjacency matrix.

TODO: Graph representation maths and description.

Model selection

ECFP model

Based on the prior knowledge encoded in Equation 1, it is reasonable to assume that certain atomic environments have a linear relationship to $\log X_{cmc}$. It therefore seems justified to apply a linear model to the ECFP fingerprints described in Equation 4. However, the issues of the large feature vector size and multicollinearity must be addressed; a naïve fit using ordinary least squares (OLS) would likely produce poor results. To that end, a process of *feature selection* was applied, whereby a subset of the atomic environments were selected for use in the model. There are several approaches to feature selection;[10] here, we chose an approach based on *regularisation*.

In this approach, we include a term in the loss function that depends on the norm of the learned parameters, or 'weights', $\vec{w}$. The two types of constraints considered in this paper are ℓ_1 and ℓ_2 regularisation, which correspond to the inclusion of ℓ_1 and ℓ_2 norms, respectively. By combining ℓ_1 regularisation with the least squares regressor, we obtain the least absolute shrinkage and selection operator (LASSO):[11]

$$\min_{\vec{w}} \frac{1}{2n_{\text{samples}}} \left\|\mathbf{X}\vec{w} - \vec{y}\right\|_2^2 + \alpha \left\|\vec{w}\right\|_1, \tag{5}$$

where n_{samples} is the number of training samples; $\mathbf{X}$ are the training data feature vectors, stacked row-wise into a matrix; $\vec{y}$ are the true values of $\log X_{cmc}$; and α is a user-defined hyperparameter describing the degree of regularisation ($\alpha \geq 0$). By imposing the ℓ_1 penalty, the model is biased towards learning a *sparse* weight vector: many of its elements will be negligible. The corresponding features can be removed from the representation.

However, LASSO has two major flaws that make it inappropriate for the task of ECFP regression:

- The number of unique atomic environments is greater than the size of the training data, but LASSO will select at most n_{samples} features.[12] Therefore, some important environments may still be excluded.

- LASSO tends to select only one of a group of highly correlated variables, when there is no reason why that particular one should be prioritised.[13] This is undesirable because we want to include both large and small atomic environments, despite their large correlation, and we might be misled into thinking that some highly correlated environments have no effect on CMC.

ElasticNet addresses these issues by imposing an additional ℓ_2 penalty:[13]

$$\min_{\vec{w}} \frac{1}{2n_{\text{samples}}} \left\|\mathbf{X}\vec{w} - \vec{y}\right\|_2^2 + \alpha\rho \left\|\vec{w}\right\|_1 + \frac{\alpha(1-\rho)}{2} \left\|\vec{w}\right\|_2^2, \tag{6}$$

where ρ is a user-defined hyperparameter controlling the proportions of the regularisation terms. This removes the hard limit on the number of features that can be selected and also exhibits the ‘grouping effect’, whereby features with high correlation tend to be assigned similar weights.

Because of these advantages, an ElasticNet linear regression model was applied to predict $\log X_{cmc}$ using the ECFP features. Both hyperparameters, α and ρ, must be defined when training the model. In order to select the best values, k-fold cross-validation was employed: the training data were partitioned into k subsets of roughly equal size. k models were trained using $k-1$ subsets, and the final subset was used for validation. The average mean-squared error of the k models, evaluated on their respective validation subsets, is the model’s final score. This routine was applied for a range of α and ρ combinations, and the lowest scoring combination was used. The hyperparameter search space is defined in the Supplementary Information.

Molecular graph model

TODO:

- Graph neural network model description.

- Hyperband description.

Results

TODO: Performance metrics, interpretations, parity plots(?).

Acknowledgement

Unsure whether to acknowledge Andrew’s specific contribution (data entry for the NIST anionics) or just leave him in the authors, as-is.

Supporting Information Available

Source code for featurisation and model training, graph neural network logs and metrics for hyperparameter optimisation and final training, and individual model predictions.

References

(1) Klevens, H. B. Structure and Aggregation in Dilate Solution of Surface Active Agents. *Journal of the American Oil Chemists Society* **1953**, *30*, 74–80

.

(2) Gaudin, T.; Rotureau, P.; Pezron, I.; Fayet, G. New QSPR Models to Predict the Critical Micelle Concentration of Sugar-Based Surfactants. *Industrial & Engineering Chemistry Research* **2016**, *55*, 11716–11726

.

(3) Jakobtorweihen, S.; Yordanova, D.; Smirnova, I. Predicting Critical Micelle Concentrations with Molecular Dynamics Simulations and COSMOmic. *Chemie Ingenieur Technik* **2017**, *89*, 1288–1296

.

(4) Mattei, M.; Kontogeorgis, G. M.; Gani, R. Modeling of the Critical Micelle Concentration (CMC) of Nonionic Surfactants with an Extended Group-Contribution Method. *Industrial & Engineering Chemistry Research* **2013**, *52*, 12236–12246

.

(5) Qin, S.; Jin, T.; Van Lehn, R. C.; Zavala, V. M. Predicting Critical Micelle Concentrations for Surfactants Using Graph Convolutional Neural Networks. *The Journal of Physical Chemistry B* **2021**, *125*, 10610–10620

.

(6) Faber, F.; Lindmaa, A.; von Lilienfeld, O. A.; Armiento, R. Crystal Structure Representations for Machine Learning Models of Formation Energies. *International Journal of Quantum Chemistry* **2015**, *115*, 1094–1101

.

(7) Himanen, L.; Jäger, M. O. J.; Morooka, E. V.; Federici Canova, F.; Ranawat, Y. S.; Gao, D. Z.; Rinke, P.; Foster, A. S. DScribe: Library of Descriptors for Machine Learning in Materials Science. *Computer Physics Communications* **2020**, *247*, 106949

.

(8) Chen, C.; Ye, W.; Zuo, Y.; Zheng, C.; Ong, S. P. Graph Networks as a Universal Machine Learning Framework for Molecules and Crystals. *Chemistry of Materials* **2019**, *31*, 3564–3572

.

(9) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling* **2010**, *50*, 742–754

.

(10) Li, J.; Cheng, K.; Wang, S.; Morstatter, F.; Trevino, R. P.; Tang, J.; Liu, H. Feature Selection: A Data Perspective. *ACM Computing Surveys* **2017**, *50*, 94:1–94:45

.

(11) Tibshirani, R. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* **1996**, *58*, 267–288

.

(12) Efron, B.; Hastie, T.; Johnstone, I.; Tibshirani, R. Least Angle Regression. *The Annals of Statistics* **2004**, *32*, 407–499

.

(13) Zou, H.; Hastie, T. Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **2005**, *67*, 301–320
.

TOC Graphic

I'm looking for ideas for the TOC graphical entry – common practice seems to be to draw a really abstract picture of a molecule going through a network, but that doesn't seem like the best thing to do here because I'm specifically comparing against other models.