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:

$$\log X_{cmc} = A - Bn_c, \tag{1}$$

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 was used to fit the models, whilst test data was 'locked away' until it had been decided that the model was optimised, and the performance metrics on the test data was used for comparison. For some models, the training data was further split into optimisation and validation subsets; the optimisation data was used when calculating the loss function during model fitting and the validation data was 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	98			
	Test	12			

We can generalise further and skip the classification by applying a similar linear model to the headgroup, based on the number of and type of chemical groups that constitute it.

In this approach, we split the molecule into *atomic environments* up to a certain radius, r: each subgraph is centred on an atom and extends r steps along connecting bonds. Now, a small 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. Tail group branching is also considered, to an extent: branch points in a carbon chain are distinguished from the rest, as they constitute a CH subgraph, rather than a CH_2.

Still, this approach relies on a discrete representation of the surfactant and, to ensure a good fit, all subgraphs of interest must be well represented in the dataset. Neural network approaches that work on molecular graphs are capable of learning a continuous representation of surfactants. They have been explored previously for CMC prediction and demonstrated promising results. However, although the continuous representation ensures that inferences can be made even for molecules with chemistry that is completely different from the training data, neural networks cannot reliably extrapolate in this way. In the case of the ECFP model, the discretisation gives a clear indicator to the researcher as to whether the model can be reliably applied: if a molecule contains subgraphs that do not appear in the training data, its prediction is less reliable. It is difficult to assess the same for a neural network.

Results

This is a test:.[2]

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

.

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.