Predicting sensor drift using neural networks

Abstract

Physical changes to gas sensors result in persistent alterations to their response properties. The unpredictable dynamics challenge the robustness of odor classifiers. To support the capabilities of the classifier to generalize to future settings, we designed a model which forms a latent representation of current classification context. Our neural network model has two parts: (1) a recurrent context layer encodes classification-relevant properties of previously seen data, and (2) a feedforward layer integrates the context with the current odor stimulus to generate a odor class prediction. The network performance is compared to a benchmark method on a sensor drift dataset.

Introduction

A challenge in gas odor recognition is that the response properties of sensors change over time, referred to as sensor drift. Drift occurs on slow and fast timescales. In order to make robust accurate predictions, a classification system must adapt to the sensor dynamics.

Researchers have created a dataset exemplifying sensor drift (Vergara et al., 2012) formatted into ten sequential classification settings, or *batches*. The data consists of recordings from 16 metal-oxide based gas sensors taken over a 36-month period. The data measures responses to six gases, ammonia, acetaldehyde, acetone, ethylene, ethanol, and toluene, which were presented without order and at variable concentration. Contributing factors to sensor drift included unrelated gases presented to the sensors between sessions, as well as time delays between batches. The features preprocessed from the timeforms are the raw and normalized steady-state features and the exponential moving average of the increasing and decaying transients taken at three different alpha values. Details about data collection and preprocessing are available in the original publication.

The identification task in the dataset is made challenging due to variable concentrations of odor as well as sensor drift contributions from chemical interference of ambient interferents and analytes. These constitute only part of the “hard problem” of olfaction in the wild (Imam and Cleland 2019). Unconstrained settings do not permit the cleaning of the sensor device between detection periods, and odors are not only variable in concentration, but also variable chemical makeup. The general problem of generalizing to new settings is a large obstacle to the translation of lab results into clinical or industrial applications (Marco 2012). In essence, even if a model performed well at classifying odors in one setting, it may fail dramatically when extrapolating outside of that setting. For this reason, our model is trained explicitly to generalize to future data.

Artificial neural networks (ANNs) are not new in the area of odor identification (Pardo and Sberveglieri). [more detail]

Methods

Sequence Modeling

The baseline setting for odor classification is to identify a target odor given evoked sensor readout data. In our method, we additionally supply a context sequence of odor samples representative of previous batches. So, our problem statement becomes to identify a target odor in batch *T* given evoked sensor readout data, along with additional historical provided from batches up to and including batch *T*-1.

The historical data is produced as follows: Suppose the unidentified target odor belongs to batch *T*. We first randomly select a context start batch *S* from 1..*T*-1. Then, for each batch in *S*..*T*-1, we construct a representative sample of odors from that batch.

To construct a representative sample of odors from batch *B*, *k* sensor responses are sampled for each class in *B*. The responses corresponding to each class are concatenated in order into a single representative sample vector representing the batch *B*.

Loss Function

The output of the model is a prediction vector with dimension equal to the number of odor classes. The values of the prediction vector are taken to be confidences that the target odor belongs to the corresponding class. Therefore, cross-entropy loss is used to evaluate the models during training.

It is important to note how training and testing data were separated. If a model is meant to be tested on batch *R*, then during training, it never is presented data from batch *R*. Instead, data sequences are constructed using only the batches 1...*R-1*. That is, *T* is sampled from 1..*R*-1.

Neural Network Architecture: The Context Model

The architecture consists of a sequence-processing context layer, a odor-processing skill layer, and a combining decision layer. All hidden layers use rectified linear activation functions.

The context layer processes the representative sample vector for each batch in the history sequence. In our work, the context layer was an LSTM with 10 units. After integrating all of the sequence, the context layer holds context for the current decision.

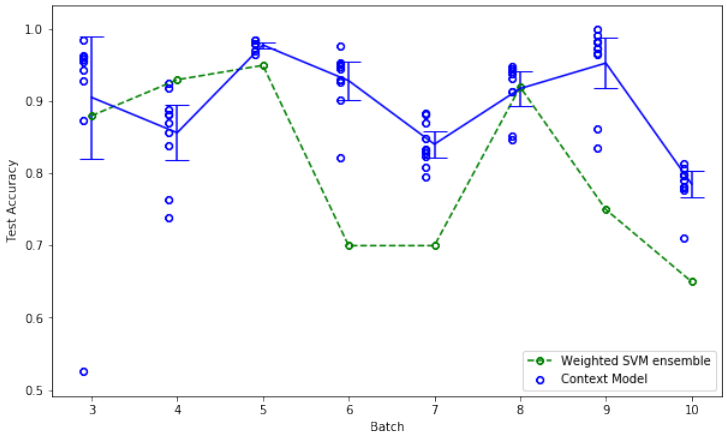
The skill layer is a feedforward, 50-unit hidden layer which accepts the target odor features as input.

The decision layer is a feedforward, 20-unit hidden layer which accepts both the context vector from the context layer and the output of the skill layer as inputs.

The network was trained end-to-end using stochastic gradient descent.

Results

The network appeared to perform well in comparison to previously reported results using a weighted ensemble model of SVMs (Vergara et al., 2012). Figure 1 shows testing accuracy on batches 3-10 of ten training runs of our network on each batch.



Conclusions

The sequence model outperformed the ensemble model.

Training a model for sequence prediction lets a model learn the temporal structure of the data, enhancing its ability to learn.