Molecular Structure Prediction Using Infrared Spectra

Category: Physical Sciences, CS 229: Fall 2017

Michael Stephen Chen (misch), Sophia Chen (schen10), Yanbing Zhu (yanbingz)

December 15th, 2017

1 Introduction

Chemists and other researchers often resort to spectroscopic methods, making use of matter-light interactions, to identify a sample's molecular composition. A given molecule's structure, including its arrangement of atoms in 3D space and interconnectivity, is manifested as a unique spectral pattern that serves the function of a molecular fingerprint. The integration of spectral clues to help narrow down a sample's identity is not systematic, often attributed to "chemical intuition", and can be prone to error. For our applications project, we trained several machine learning models on a collection of different IR spectra in order to correctly classify for the structure of the corresponding organic molecule. More specifically, given the spectral intensities at a standard set of frequencies as our training inputs, we looked to classify the molecule's functional groups, ubiquitous structural motifs that have well-defined well defined physical and chemical properties, using logistic regression, k-means clustering, and a feedforward neural network.

2 Related Work

Various machine learning methods have been applied to a range of different spectroscopy studies over the years. Ellis et al. applied a multiple linear regression method to successfully classify the different types of muscle foods (beef, lamb, pork, chicken, turkey) based on the corresponding IR spectra. Additionally, they were able to accurately classify and authenticate different kinds of meat with a similarly trained model. Howley et al. classified chemical samples containing acetaminophen using SVMs and k-means with a PCA-reduced Raman spectra feature set [2]. You et al. trained an SVM on near-infrared (NIR) spectra for the identification of oxytetracycline powder [3]. Martelo-Videl et al. made use of Vis-NIR spectra to train an artificial neural network (ANN) to classify different wines [4]. Generally speaking, applying Soft Independent Modeling of Class Analogy (SIMCA) and Partial Least Squares (PLS) on data, with the features having been reduced in dimensionality using PCA, are both common chemometric methods for classification [5].

Despite the booming interest in applying machine learning classification methods to spectral data, most of literature to date seeks to classify for a highly specific compound or set of compounds (e.g. muscle foods, wine, etc.). There is little published work regarding the more challenging problem of predicting for the general molecular structure of a sample based on spectral data. For our project we took a step in that direction by using some of the above machine learning methods trained on IR spectra to identify an organic molecule's functional groups.

3 Dataset and Features

The data used was scraped from the NIST Chemistry Webbook, which contains IR spectra for 16k compounds [6]. The IR spectra, the 3D spatial data files (SDF) that provide the molecular geometry, and the International Chemical Identifier files for organic molecules with up to 9 carbons were downloaded using a modified BeautifulSoup script [7]. In the end, we had a relatively small set of 1685 molecules to train on. We then parsed the files to extract the spectral data into array form. However, we realized that the data we downloaded was not consistent across all spectra. Some of the spectra used wavenumbers (cm^{-1}) as the x-axis while others used wavelengths (μm). We also realized some of the spectra we downloaded had transmittance while others had absorbance. Thus, before we could run any machine learning methods, we had to convert every spectrum into absorbance spectra with wavenumbers as the x-axis and standardize the intensities by dividing by the maximum absorbance value to get a ratio. We also realized that the resolution of the spectra varied greatly from sample to sample. To address resolution inconsistencies, we decided to standardize the spectra restricting the range of values considered to 400-4000 wavenumbers. The features for our models were the normalized peak intensities for bins of wavenumbers. The exact resolution of our features was cross-validated to get 1000 bins (Figure 1). The intensities for the standardized set of wavenumbers were determined via linear interpolation between the two closest points.

Additionally, we parsed the 3D SDF files and used the extracted information regarding the sample molecule's atoms and interconnectivity to label the molecules with the functional groups they were made up of. Labels with less than 10 molecules were discarded since we would not have enough training or testing samples to properly identify those labels. The 13 functional groups we used to label our data were alkanes, alkenes, alkynes, alcohols, amines, nitriles, aromatics, alkyl halides, esters, ketones, aldehydes, carboxylic acids, and acyl halides. We also included a label for carbonyls, which grouped together esters, ketones, aldehydes, and carboxylic acids, for logistic regression and k-means [8].

For logistic regression and k-means, we only distinguished between carbonyls, alkenes, and alcohols, so our sample size was 1384 samples. We further split the samples into training and testing, with 1245 training samples and 139 testing samples (approximately 10% of the samples used for testing). We ran the neural network on all 13 functional groups, with 1516 training samples (of which 379 were set aside for k-fold validation) and 169 testing samples.

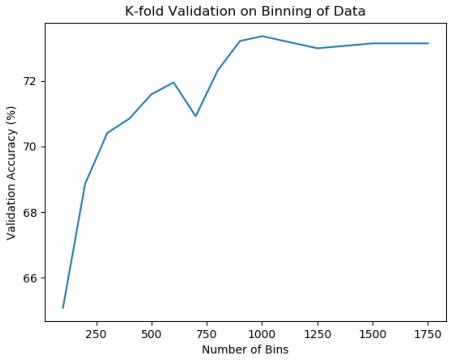


Figure 1: Number of bins for data

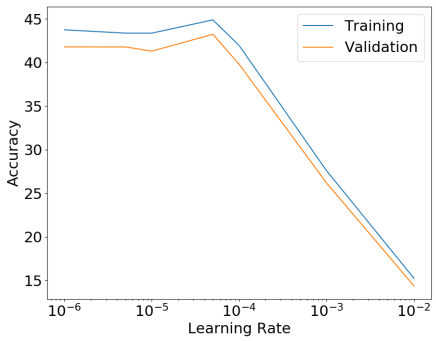


Figure 2: Training and Validation Accuracy for Logistic Regression Learning Rate

4 Methods and Experiments

4.1 Logistic Regression

We applied a One-vs-Rest logistic regression classifier using a stochastic gradient descent solver to classify carbonyls, alkenes, and alcohols, and calculated the training and testing errors using Scikit-learn methods [9].

Logistic regression can be conceptualized by modeling our class probabilities as sigmoids and taking the most likely estimate. In doing so we can arrive at the following stochastic gradient descent update:

$$\theta_j := \theta_j + \alpha \left(y^{(i)} - h_\theta \left(x^{(i)} \right) x_j^{(i)} \right) \tag{1}$$

For our implementation we decided upon a learning rate $\alpha = 5 \times 10^{-5}$ via 4-fold crossvalidation. A plot of accuracy vs. α is provided in Figure 2.

4.2 k-means

We used k-means to cluster the IR spectra of carbonyls, alkenes, and alcohols into three different, distinct functional groups. To do so, we first ran Scikit-learn's k-means algorithm directly on the samples [9]. Three cluster centroids were chosen for the above three functional groups and the initial values were randomly generated. The spectra for each cluster centroid was taken to be the mean value at each wavelength for all the samples assigned to that center and iterated until convergence. We also used principal component analysis (PCA) to reduce the amount of components of the samples to three components that would capture the three main groups. We then ran k-means again on the reduced components. Although k-means is meant to be an unsupervised method, since we had the luxury of labeled data, we were able to calculate a training and testing error based on the predictions that k-means made.

4.3 Neural Network

Neural networks, generally speaking, are a type of nonlinear classifier. A neural network effectively models the architecture of a brain, with nodes as our neurons and edges between the nodes that act like synapses. The leftmost nodes in the diagram represent the inputs and the rightmost represent the outputs of neural network classifier. The edges denote the transfer of information outputted from one node to another. Thus a neural network is effectively compositions of composite functions, thus giving it its nonlinear classification abilities. We designed a two-layer feedforward neural network to classify all 13 mentioned functional groups using Tensorflow [10]. To determine the hyperparameters for our neural network, which included the number of hidden nodes per layer and the learning rate of our stochastic gradient descent updates, we ran four-fold validation (validation sample size of 379 samples) on the hyperparameters for our neural network. Once our parameters were set, the input to our neural network was all 1516 training samples. Our activation

function for the hidden layers was the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$, and our activation function for the output layer was the softmax function. Our output was a one-hot vector of individual functional groups, as well as all possible combinations of functional groups (Figure 4). We ran the training for 1000 epochs, and calculated the accuracy on the test set at the end of the 1000 epochs.

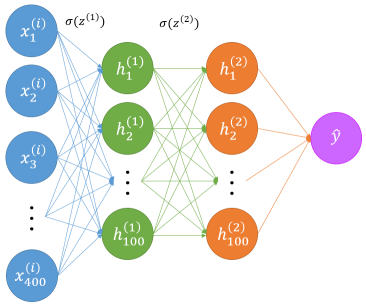


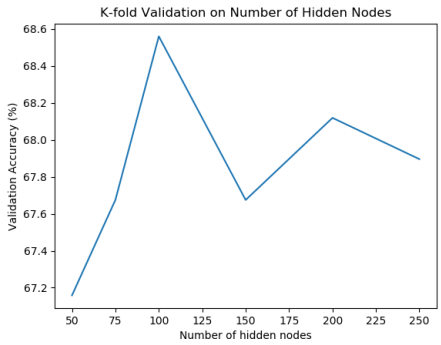
Figure 3: Neural Network Configuration. $x^{(i)}$ is sample i, $h^{(1)}$ is the first hidden layer, $h^{(2)}$ is the second hidden layer, and the output is $\hat{y}$

alkanes	alkenes	...	acyl halide	...	alkane and alcohol	alkane and amine	...	(combinations of 2 groups)	...	alkane and alcohol and amine	...	(combinations of 3 groups)	...	(other possible combinations)	...	
[1	0	...	0	...	0	0	...	0	...	0	...	0	...	0	...]	one-hot vector for molecules with alkanes
[0	0	...	0	...	1	0	...	0	...	0	...	0	...	0	...]	one-hot vector for molecules with both alkanes and alcohols

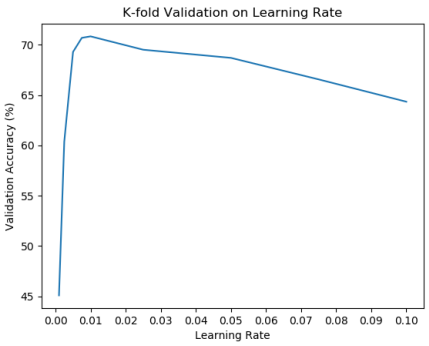
Figure 4: One-hot label examples. Note that any group or combination of groups that has less than 10 molecules under its label is not included

5 Results

We ran four-fold validation on our hyperparameters for the neural network and concluded that the optimal parameters for our neural network were 100 nodes per hidden layer (Figure 5a) and a learning rate of 0.01 (Figure 5b).



(a) Number of hidden nodes for neural network



(b) Learning rate of weight updates for neural network

Figure 5: Plots of k-fold validation accuracy

Our overall training and testing errors can be found in Table 1.

Model	Training Error	Training Accuracy	Testing Error	Testing Accuracy	Training/Testing Samples
Logistic Regression	41.54%	58.46%	42.14%	57.86%	1245/139 samples
k-means	43.62%	56.38%	46.00%	54.00%	1245/139 samples
k-means w/PCA	43.41%	56.59	45.99%	54.01%	1245/139 samples
Neural Network	9.30%	90.70%	20.53%	79.47%	1516/169 samples

Table 1: Training and Testing Errors for Models

Functional Group	Training Error	Testing Error
Carbonyls	8.53%	14.28%
Alkenes	50.67%	53.96%
Alcohols	45.40%	43.47%

Table 2: Training and Testing Errors for Individual k-means Clusters

k-means and logistic regression both performed with around 50% error, indicating that both methods were fairly inconclusive for predicting the three functional groups. However, looking at the separate training and testing errors for each functional group for k-means (Table 2), we can see that the errors for classifying carbonyls were actually very low, but the errors for alkenes and alcohols were quite high, thus increasing the overall error. We can also see that one of the spectra that k-means converged to seems to have a mix of alkene and alcohol peaks (Figure 6), which accounts for its inability to properly classify alcohols and alkenes. PCA did decrease the error for k-means, but almost by nothing.

Our neural network was able to classify 13 different functional groups and all their possible combinations relatively successfully (Figure 7).

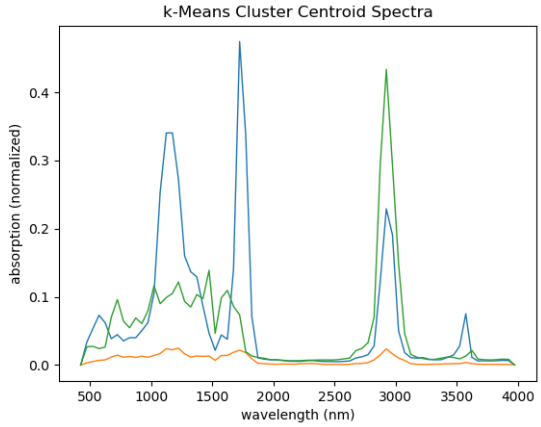


Figure 6: Spectra for k-means cluster centroids after running on alcohols, alkenes, and carbonyl groups.

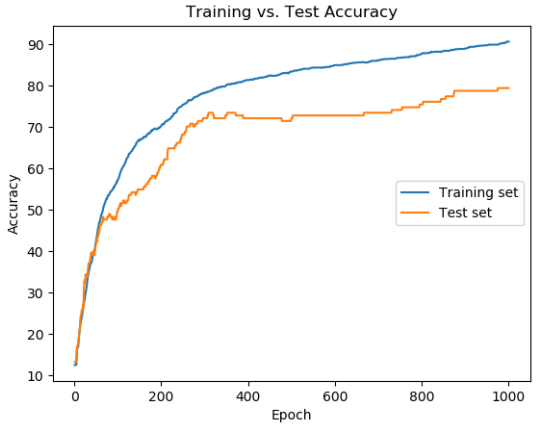


Figure 7: Training and test accuracy over epochs of neural network training

6 Discussion

In attempting to classify a molecule's functional groups based on its IR spectrum, we conducted a survey of methods and found our neural network gave the highest performance. We expected this result given the nonlinear nature of the decision boundary, as typified by the overlapping of characteristic peaks amongst different functional groups. Thus, logistic regression did poorly, doing a little better than guessing the three different classifications. The neural network, on the other hand, classifies the functional groups decently well, with only 20.53% testing error, and is able to get the important peaks while ignoring the insignificant ones.

We also hypothesized that k-means would be able to cluster the molecules based on the characteristic peaks of each functional group. However, because characteristic peaks are often located in similar spots, k-means is likely to only be

able to distinguish between functional groups with very different locations. The other functional groups besides carbonyls, alkenes, and alcohols had peaks that were too similar to other functional groups. Even with these supposedly dissimilar peaks, k-means was unable to clearly cluster into spectra for the three different functional groups. Individually, it was able to classify carbonyls very well, but for the other two groups, it was unable to distinctly classify them, probably because some compounds had both alkene and alcohol functional groups and k-means put them together into one spectrum. Regardless of the number of groups, one cluster (orange) was always suppressed in amplitude, presumably picking up the less distinct spectra (Figure 6). Initially, we also hypothesized using PCA for dimensionality reduction would additionally help improve accuracies, given that certain frequencies appear to not be specific to any functional group. However, in applying PCA we found there to be only slight accuracy improvements. With PCA, we were able to slightly lower our error after reducing to four components. We suspect that the overlap in functional group peaks would also affect PCA in the same way it affected k-means.

From Figure 7, we can see that we are overfitting the data to some extent as the testing error tapers off a bit while the training error systematically increases. We attribute this overfitting mainly to the high-dimensional nature of the features we are using (1000 different frequency intensities). Neural networks in general, given all of the parameters involved, tend to be prone to overfitting. To combat overfitting, we tried to both decrease the resolution of our features and also reduce the number of hidden layer nodes in our neural network. To find the optimal bias vs. variance number for these hyper parameters, we performed crossvalidation over a range of values for the respective parameters and the results are presented in Figures 1 and 5a.

7 Conclusion

Actual sample composition determination usually involves integrating clues from multiple sources, and analogously we would extend our models to account for a combination of spectral features in addition to IR (e.g. UV-Vis, NMR, Mass-spec, etc.). This would help identify functional groups better, and we could essentially run the same methods on this data. Ideally, we would extend the problem of classifying a molecule based on common functional groups to the regression problem of determining the molecule's exact elemental composition and 3D structure. This would take a lot more work, and currently there are not a lot of methods for reconstructing 3D structures besides human reconstruction. We could also look into improving our neural network by using a more complex one, such as a convolutional neural network. However, in order to do so, we would need more computing resources and a better understanding of how to use various neural network libraries, such as Tensorflow.

8 Contributions

All of us contributed equally to the writing of this report. While we decided on most of our preprocessing and machine learning methods together, we split up the work.

Sophia worked on parsing and processing the IR spectra, including converting it into absorption spectra and standardizing the x-axis. She also worked on the k-means algorithm and getting the errors for it. Sophia wrote the neural network and ran cross-validation on number of hidden nodes.

Yanbing worked on the BeautifulSoup script for downloading the data from NIST and running preliminary PCA results building off of k-means.

Michael wrote the script for labeling the spectra by functional groups, standardizing the data, and running SVM and linear regression. He also optimized parameters for the logistic regression (learning rate) and neural network (learning rate and resolution of features) implementations using cross-validation.

References

[1] Ellis, David I, et al. "Rapid identification of closely related muscle foods by vibrational spectroscopy and machine learning." *Analyst* 130.12 (2005):1648. Web.

[2] Howley, Tom, et al. "The effect of principal component analysis on machine learning accuracy with high-dimensional spectral data." *Knowledge-based Systems* 19.5 (2006):363-370. Web.

[3] You, T., et al. "Support vector regression for determination of component of compound oxytetracycline powder on near-infrared spectroscopy." Analytical Biochemistry 355 (2006) 1-7. Web.

[4] Martelo-Vidal, M. J. "Application of artificial neural networks coupled to UV–VIS–NIR spectroscopy for the rapid quantification of wine compounds in aqueous mixtures." CyTA - Journal of Food 13(1), 32-39 (2015). Web.

[5] Madden, M. G., T. Howley. "A machine learning application for classification of chemical spectra." Proc. of 28th SGAI International Conference, Cambridge, UK, (2008). Web.

[6] Stein, S.E., NIST Mass Spec Data Center. "Infrared Spectra" in NIST Chemistry WebBook, NIST Standard Reference Database Number 69, Eds. P.J. Linstrom and W.G. Mallard, National Institute of Standards and Technology, Gaithersburg MD, 20899, doi:10.18434/T4D303, (retrieved December 5, 2017).

[7] Swain, M. Github (2015). `https://gist.github.com/mcs07/48fcfc0f072e5f45dcaa`

[8] Merlic, C. "Table of IR Absorptions." UCLA, Los Angeles, CA, (2000). `https://webspectra.chem.ucla.edu/irtable.html`

[9] Pedregosa, F., et al. "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research* 12 (2011):2825-2830. Web.

[10] Abadi, M., et al. "Tensorflow: Large-scale machine learning on heterogeneous systems", 2015. Software available from tensorflow.org. Web.