Automated Crystal Structure Identification from X-ray Diffraction Patterns

Rohit Prasanna (rohitpr) and Luca Bertoluzzi (bertoluz)
CS229: Final Report

1 Introduction

X-ray diffraction is a commonly used experimental technique for identifying the structure of crystalline materials[3]. A crystalline material has its atoms arranged in periodic lattices with long-range symmetry. As a result, it scatters a beam of X-rays in well-defined patterns. Analysis of an X-ray diffraction pattern produced by a material yields extensive information about the structure of the material, in particular, about the kind of periodic lattice its atoms are arranged in.

Currently, basic analysis of X-ray diffraction patterns is often performed manually, often requiring human input and some guesswork at what the structure of an unknown material is likely to be. This makes it hard to use X-ray diffraction data in automated analysis of large datasets. Automating the process of structure determination from X-ray patterns would be a step toward high-throughput computational searches for materials with desirable properties - a goal that the materials science community has been moving to in recent years as available computational power has increased.

This project aims to automatically classify the crystal structure of an unknown material. There are seven possible basic crystal systems (also known as Bravais lattices) that a crystalline material can adopt, and the measured XRD pattern of the material contains sufficient information to say which crystal system the material belongs to. The input to our program is an XRD pattern that consists of scattered X-ray intensity as a function of a variable called the reciprocal lattice vector (denoted q). We discretize this pattern by slicing up the q-space into a variable number of pieces and use the intensity at each q as a feature. The output is a classification of the given XRD pattern into one of the seven basic crystal systems. We implement a Naive Bayes classifier and a neural network to perform the classification and compare their performance.

2 Related Work

The method used most often to analyze XRD patterns is an manual process that starts with a guess at what the crystal structure is likely to be, then simulates the XRD pattern that would result from the guessed structure, and iteratively improves the structure in order to match the simulated XRD pattern to the measured one. There exist programs to aid in this process, such as GSAS-II[5], which contains powerful tools to refine structures derived from experimental XRD data. However, while this and other tools work very well in performing sophisticated analysis beyond just classifying a material's crystal structure, they often require specialized knowledge and expertise to use.

3 Dataset and Features

The training and test data were generated by downloading approximately 12,000 known crystal structures from the Inorganic Crystal Structure Database[1]. An open-source program, Platon[2], was used to simulate a diffraction pattern for each structure using first principles physics. Homebuilt code was used to identify the crystal system for each of the known structures, and each simulated XRD pattern was tagged with this known output.

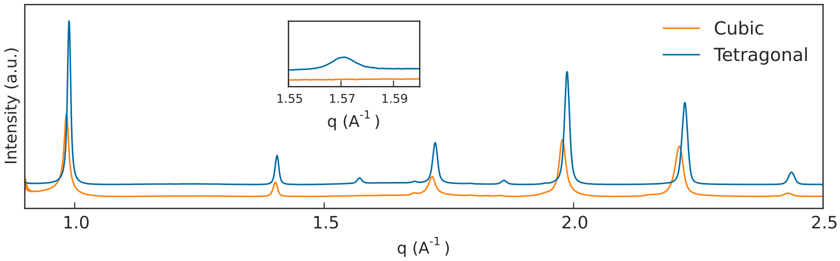

Figure 1: Two examples of XRD patterns belonging to two different crystal systems, cubic and tetragonal. The zoomed in view in the inset highlights a distinguishing feature in the tetragonal pattern that is not present in the cubic.

Each training example is a pattern of X-ray intensity (I) as a function of what is called a reciprocal lattice vector (q) over a fixed range of q, accompanied by an output label indicating which crystal system the pattern corresponds to. The model we use describes each such pattern by discretizing it in q to generate a vector of dimension equal to the number of discrete q-steps chosen. Figure 1 shows two examples of XRD patterns.

4 Methods

4.1 Naive Bayes classifier

We use a supervised learning setting with a Naive Bayes classifier. To implement this method, we discretize each XRD pattern in both axes - such that each pattern is represented as a vector of dimension equal to the number of discrete steps chosen in q-space. The intensity at each q is modelled as a multinomial distribution over a finite number of steps. Using these feature vectors, we implement a Naive Bayes classifier to model probabilities of a new pattern coming from each of the seven crystal systems. The basic assumption that the Naive Bayes method makes is that separate features in the input are conditionally independent of one another, given the output - that is, the probability of one input feature having a certain value has no bearing on the probability of a different feature, even when the correct output corresponding to that input vector is known.

This simple (but often inaccurate) assumption allows one to build a model for the conditional probability of an input vector as a simple product of the probabilities of individual features of the input.

$$p(x_1, x_2|y) = p(x_1|y) \times p(x_2|y) \tag{1}$$

Taking the individual terms on the right as parameters, the model then computes the posterior probability p(y|x) using Bayes' rule. The optimal values of the parameters of the model, p($x_i|y$), p(y) are computed by empirically calculating the relevant probabilities from the training dataset. The class with the highest posterior probability, that is - argmax$_i$p(y$_i|x$) is the classification that the model returns for a new input vector x.

4.2 Neural Network

Our second method for classifying unknown XRD patterns uses a neural network. The input layer consists of an XRD pattern discretized in q-space, using a variable number of steps, such that the number of input features is systematically varied to study its influence on the results. The network contains either one or two hidden layers, which use the sigmoid function for activation. The output layer consists of seven nodes, each of which corresponds to one of the possible crystal systems that the network has to classify an unknown pattern into. We train the network on approximately 11000 XRD patterns pre-labelled with the known output, by minimizing the cross-entropy loss averaged over all the training examples:

$$J = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=0}^{6} y_k^{(i)} \log a_2^{(i)} \tag{2}$$

where the sums are over all m training examples and all seven possible outputs, $y^{(i)}$ represents the ground truth, and $a_2^{(i)}$ represents the prediction from the model.

5 Experiments and Results

We trained the Naive Bayes model described above on a set of 10,890 XRD patterns, sampled in 4000 angle intervals (i.e 4000 features). We tested our model on a set of 1210 examples (10% of the total number of XRD patterns we generated). The Naive Bayes predictor performs poorly, with a mere 31% accuracy at predicting the correct crystal system for a given XRD pattern.

To check the performance of our model, we perform error analyses based on the number of training examples and the number of features in the training and test matrices, as shown in Figure 2. Our analysis reveals that above 800 features, the error of our model is approximately constant at 69%, while it increases for a lower number of features. Similarly, the error of the Bayesian model increases as the number of training examples decreases and saturates above 1000 training examples. Notably, both the training and test errors remain high and in the same range.

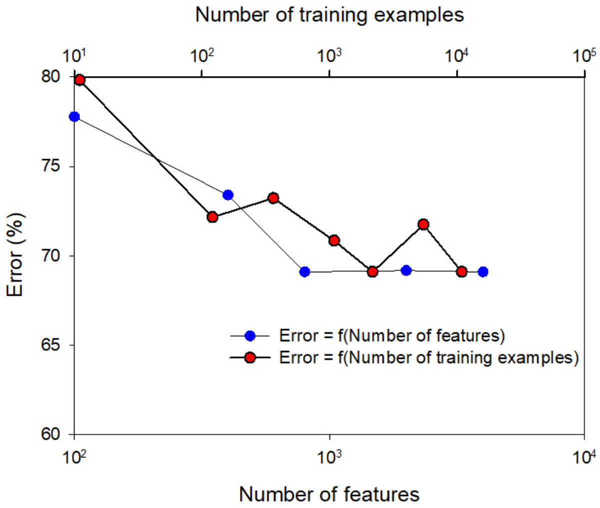

Figure 2: Error analysis on our Bayesian model as a function of the number of training examples and number of features.

Looking into the predicted probabilities of a new pattern being of a particular crystal system shows that for a given pattern, the probabilities associated with each of the seven crystal systems do not differ by very much. In other words, the model cannot clearly distinguish between all the Bravais lattices based on a distribution of intensities of XRD patterns under the Naive Bayes assumption.

This simple error analysis indicates that it is unlikely that expanding the size of the training dataset or the number of features used will successfully decrease the model's error. Our conclusion is that a Bayesian model is not adapted for the analysis of XRD spectra. The pattern produced by a given crystal system typically contains peaks whose positions are correlated to one another in ways that correspond to what crystal system the material belongs to. This violates the main assumption of the Bayes classifier, which requires that input features are conditionally independent, given the output. This is likely the main reason for poor performance of the Naive Bayes model for this problem.

The neural network performs significantly better than the Naive Bayes classifier. A training accuracy of close to 100% is achieved within 15 iterations over the training dataset in a batch gradient descent optimization. The test accuracy of 56% achieved with a single hidden layer and no regularization is significantly better than that produced by Naive Bayes and is much higher than would be produced by random guessing (14.3%) among the seven possible outputs. However, it is still far lower than 100%. Noting that the training accuracy is high while the test accuracy is much lower, we hypothesize that the model is overfitting the training data and capturing trends that do not reflect underlying mechanisms and do not generalize to data outside of the training dataset. To mitigate this problem, we apply regularization by adding the squared norms of the weight matrices for the hidden and output layers, multiplied by a regularization coefficient λ, to the cost function.

$$J = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=0}^{6} y_k^{(i)} \log a_2^{(i)} + \lambda(||W^{[1]}||^2 + ||W^{[2]}||^2) \tag{3}$$

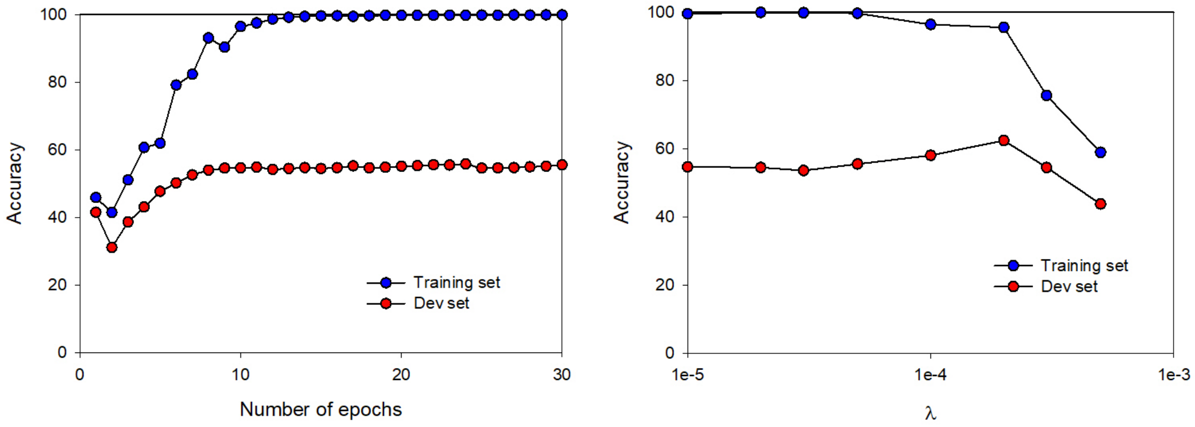

Figure 3: (a) Learning curves for the neural network, showing training and test accuracy as functions of the number of iterations (epochs) over the entire dataset. (b) Training and test accuracy of the neural network as the regularization coefficient is varied.

where $W^{[1]}$ and $W^{[2]}$ are matrices containing the weights used for the hidden and output layers. We vary the size of this coefficient λ and see that regularization does improve the test accuracy, from 56% to 62%, with the best result occurring for λ of 2×10^{-4}.

6 Conclusions and Future Work

Naive Bayes method applied to this problem attained poor success, which we expect is due to extensive correlations among different input features (i.e. presence of correlated peaks in the XRD pattern), given the output (crystal system). A neural network significantly improves upon this, achieving 62% accuracy in classifying XRD patterns in the test set. However, the neural network seems to overfit the training data extensively, resulting in high accuracy on training data but mediocre performance on test data. Future work can aim to reduce this by implementing dropout, or a random omission of some units and their connections during training. This technique has been shown to reduce the problem of overfitting in deep learning systems[4].

7 Contributions

R.P. collected the crystal structure data and simulated the patterns for use in training and test sets. Both L.B. and R.P. designed the schemes to be used for representing data. L.B. implemented the Naive Bayes method in Matlab. Both R.P. and L.B. implemented the neural network. Both R.P. and L.B. performed error analysis and diagnostics. R.P. wrote the report.

References

[1] Inorganic crystal structure database. https://icsd.fiz-karlsruhe.de/. Accessed: 2017-10-15.

[2] Platon. https://www.platonsoft.nl/xraysoft/unix/platon. Accessed: 2017-10-15.

[3] Bernard Dennis Cullity, Stuart RBD Cullity, and SR Stock. *Elements of X-ray Diffraction.* Number Sirsi) i9780201610918. 2001.

[4] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.*, 15:1929–1958, 2014.

[5] Brian H. Toby and Robert B. Von Dreele. GSAS-II: The genesis of a modern open-source all purpose crystallography software package. *J. Appl. Crystallogr.*, 46(2):544–549, 2013.