Predicting Phase of Simulated Molecules Using Radial Structural Functions

by Heejung Chung
advised by Rodrigo Freitas *(Reed Materials Computation and Theory Group)*
Category: Physical Sciences

Introduction and Related Work

Background: Radial structural functions

For the *i*th molecule in a sample, the radial structural function (RSF) at radius *r* around the molecule is

$$G^{(i)}(r) = \sum_{j=1}^{n} e^{-(d_{ij}-r)^2/2\sigma^2} \quad ,^{(2,5)} \tag{1}$$

where n is the number of neighbors, d_{ij} is the distance from molecule *i* to its *j*th neighbor, and σ is a chosen constant. $G^{(i)}(r)$ is approximately equal to the number of neighboring molecules that are distance *r* away from the *i*th molecule. For example, if the red molecule in Fig. 1 is the *i*th, then below are its approximate RSFs.

$G^{(i)}(r_1)$	$G^{(i)}(r_2)$	$G^{(i)}(r_3)$	$G^{(i)}(r_4)$
≈ 4	≈ 5	≈ 3	≈ 4

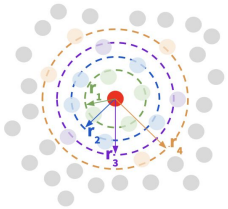

Figure 1: How to approximate RSFs

RSFs can be obtained only from molecular dynamics simulation and have been shown to be useful for determining properties of individual simulated Silicon molecules[1,5].

Background: Radial distribution function

The radial distribution function (or RDF) of a material at radius *r* is given by

$$g(r) = \lim_{\sigma \to 0} \frac{avg(G(r,\sigma))}{\rho\Omega(r,\sigma)}, \tag{2}$$

where ρ is the density of molecules in the substance and $\Omega(r,\sigma)$ is the volume of a spherical shell of average radius r and thickness of 2σ. The RDF of a sample is essentially the scaled average of RSFs over all molecules in the material. A sample's RDF is obtained experimentally, but is produced in this project artificially through simulation. One can approximate the RDF $g(r)$ from the $avg(G(r,\sigma))$ or vice versa by setting σ to some small number.

Predicting phase of simulated water molecules using RSFs

The first element of this project addresses the issue that there is no current consensus among materials science computation researchers about the best way to featurize the mictrostructure of a material. Here, it's confirmed that RSFs may be good candidates for featurizing microstructure[3], since they can be used to predict the phase of H_2O molecules.

Machine Learning Problem Setup (supervised): The input for the algorithm is a vector of RSFs for a simulated H_2O molecule. In other words, each example corresponds to a molecule, and the features are RSFs at different radii, so that the feature vector looks like

$$x^{(i)} = [G^{(i)}(0\text{ Å}), G^{(i)}(0.1\text{ Å}), \ldots\ G^{(i)}(5.9\text{ Å}), G^{(i)}(6\text{ Å})],\ x^{(i)} \in R^{60}. \tag{3}$$

The label $y^{(i)}$ for an example is +1 if the *i*th molecule is in the liquid phase (water) and –1 if it is in the solid phase (ice). I then used an SVM to predict phase of a molecule based on RSFs.

Generating RSFs using RDF

The second element of this project addresses the issue that experimental data can be expensive and difficult to obtain. If a researcher has a sample of unknown composition, the process of measuring its RDF can destroy the sample. Since data like the RDF (which only characterizes macrostructure) can be costly, it would be useful to *augment* such experimental data with what can be learned from simulated data. One way to do this is to learn how to sample RSFs (which characterize microstructure) from a probability distribution parameterized by the RDF of a material.

Machine Learning Problem Setup (unsupervised): In this project, I assumed for simplicity that RSFs at different radii are independent of each other. I wanted to learn the distribution

$$G^{(i)}(r_j) \mid g(r_j) \sim P(\theta_{g(r_j)})$$

where P is some probability distribution and $\theta_{g(r_j)}$ is a parameter of P based on the RDF at radius r_j. I found that sampling RSFs from exponential distributions generated fake RSFs that were distributed more similarly to real RSFs than fake RSFs sampled from gaussian distributions.

Dataset and Features

Predicting phase of simulated water molecules using RSFs

I ran two LAMMPS[4] molecular dynamics simulations at 250K containing 768 molecules each, one of ice and one of water, and computed the RSFs of all molecules at two timestamps (t1 and t2) from each simulation. Times t1 and t2 were far enough apart that I could assume the distribution of molecules' RSFs are independent of the timestamp they are taken from. At this point I had four sets of data, the set I_1 of ice molecules' RSFs at t1, the set I_2 of ice RSFs at t2, the set W_1 of water RSFs at t1, and the set W_2 of water RSFs at t2. Note that all examples in I_1 and I_2 are negatively labelled while all examples in W_1 and W_2 are positively labelled (as described in the **Introduction** section).

I formed my train and dev sets by setting

$$\text{Train set } T_1 = I_1 \cup W_1$$
$$\text{Dev set } D = I_2 \cup W_2.$$

The train and dev sets contain 1536 examples each. Any time I trained an SVM I whitened the training data beforehand.

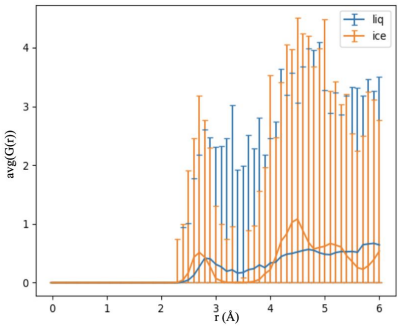

Figure 2: Mean RSF G(r) as a function of r (from set T_1, error bars show min and max G(r) at each r)

Generating RSFs using RDF

I generated a new set F_1 composed only of fake RSFs sampled from gaussian distributions. I assumed that for each $r_j \in \{0\text{Å}, 0.1\text{Å}, \ldots 5.9\text{Å}, 6\text{Å}\}$,

$$G^{(i)}(r_j) \mid g(r_j) \sim N(\mu_j, \sigma^2), \text{ where}$$
$$\mu_j = avg(G(r_j, \sigma)) = \rho\Omega(r,\sigma)g(r_j), \quad (4)$$

and σ is the same σ used to calculate all G(r) and g(r) according to equations (1) and (2). I obtained the RDFs from the previously mentioned ice and water simulations at t1. Using the above assumption, I sampled fake RSFs corresponding to 768 fake ice molecules and 768 fake water molecules (where each example of fake RSFs took the same format as equation (3)). I formed F_1 by taking the union of these two datasets.

I then similarly generated another set F_2 of RSFs (for 768 fake ice, 768 fake water molecules) sampled from exponential distributions. Here, the assumption was that for all r_j,

$$G^{(i)}(r_j) \mid g(r_j) \sim Exp(\lambda_j), \text{ where}$$
$$\lambda_j = [avg(G(r_j))]^{-1} = [\rho\Omega(r,\sigma)g(r_j))]^{-1}. \quad (5)$$

To justify these choices of parameters, note that since

$$avg(G(r_j)) = \frac{1}{n}\sum_{i=1}^{n} G^{(i)}(r_j),$$

μ_j from equation (4) and λ_j from equation (5) are the maximum likelihood estimates of the parameters for their respective distributions. The σ was chosen to be the same across all r_j, because in real life if an RDF were obtained experimentally, there wouldn't necessarily be a measurable error term on $g(r_j)$ for each r_j that could be used to compute an MLE standard deviation.

Methods

Predicting phase of simulated water molecules using RSFs

I trained an SVM with a linear kernel to predict the phase of a molecule based on its RSFs. A linear support vector machine in the separable case finds a plane such that

- all positively labelled examples fall on one side of the plane, all negatively labelled examples fall on the other,
- distances from the plane to the nearest positively labelled example(s) and nearest negatively labelled example(s) are maximized.

The objective function with regularization term C is

$$min_{\gamma,w,b} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i$$
$$\text{subject to } y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i,\ i = 1, ..., n$$
$$\text{and } \xi_i \geq 0,\ i = 1, ..., n\,.$$

Results and Discussion

Predicting phase of simulated water molecules using RSFs

As a baseline, I trained a linear SVM without regularization on T_1 and validated it using dev set D, obtaining a train accuracy of 94.5% and validation accuracy of 94.3%. Note that since my classes (+1 for liquid vs –1 for solid) were always equally represented in my data, I focused more on accuracy than precision or recall as my evaluation metric.

Though the unregularized SVM performed very well, I also plotted a validation curve (Fig 3) using 5-fold cross validation on T_1, and found that the optimal regularization term C=10.

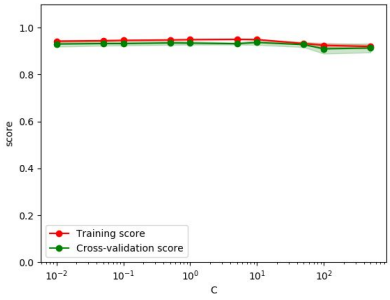

Figure 3: Validation Curve on T_1 to find C

After training a regularized (C=10) linear SVM on T_1 and validating it using dev set D, I obtained a train accuracy of 95% and validation accuracy of 94.3%. I also plotted a histogram of the decision function (Fig 4) for the dev set and found that many of the ice molecules' RSFs were concentrated relatively close to the SVM hyperplane compared to how the water molecules' RSFs were distributed.

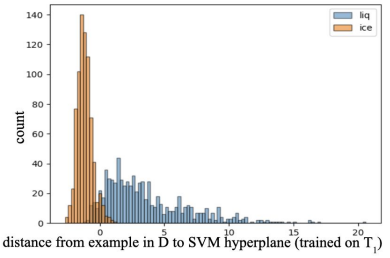

Figure 4: Histogram of Decision Function on D (trained on T_1)

Generating RSFs from RDF

I wanted a measure of how similarly distributed T_1 and F_1 were compared to T_1 and F_2. As an initial measure, I evaluated this similarity based on how well SVMs trained on F_1 and F_2 (fake RSFs) could predict phase of molecules in the dev set D (real RSFs). In the future, I may try to use a more complex metric of estimated similarity.

I first trained an SVM on F_1 and found that the accuracy when predicting phase of the dev set was 83%. When I plotted a histogram of the decision function for D, I found that the distances to the hyperplane found by F_1 were distributed very differently (Fig 5) from how they are distributed in Fig 4.

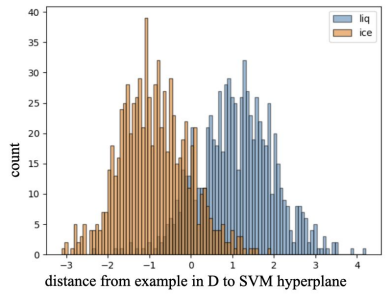

Figure 5: Histogram of Decision Function on D (trained on F_1)

This difference suggests that the hyperplane separating water and ice RSFs in F_1 is very different from the hyperplane separating RSFs in T_1.

I then trained an SVM on F_2 and found that the accuracy when predicting phase of examples in D was 89%. This already suggests that F_2 is distributed more similarly to T_1 than F_1. In addition, the histogram (Fig 6) of the corresponding decision function on D looks much more similar to Fig 4.

The improvement in accuracy and decision function histogram shape from the SVM trained on F_1 to the one trained on F_2 suggests that RSFs are more exponentially distributed (rather than gaussian).

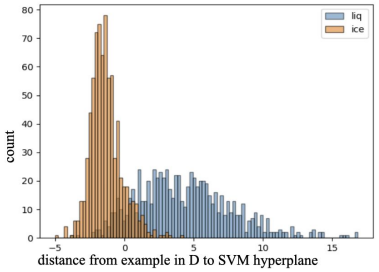

Figure 6: Histogram of Decision Function on D (trained on F_2)

Overview of results

Model	Train Accuracy	Dev Accuracy
unregularized SVM trained on real RSFs	94.5%	94.3%
SVM trained on real RSFs, C=10	95%	94.3%
SVM trained on gaussian fake RSFs (F_1)	100%	83%
SVM trained on exp. fake RSFs (F_2)	100%	89%

Conclusion and Future Work

Predicting phase of simulated water molecules using RSFs

By using an SVM to predict phase of molecules using RSFs, I confirmed that RSFs encapsulate the phase of a sample and may be promising as general featurizations of material structure. In the future, I may try to gather more data, namely RSFs from simulation of a mixed-phase sample (eg 70% water, 30% ice), and see if the SVM trained on T_1– based on the percentage of examples that are predicted to be water vs ice– can predict the composition of the sample.

I can also look at whether or not RSFs capture more complex structural properties. For example, I could run more molecular dynamics simulations of materials with certain defects, along with simulations without defects. I could then see if an SVM (or more complex machine learning algorithm if necessary) can predict whether or not a molecule is near a defect or not.

Generating RSFs from RDF

It's possible to augment experimental data (RDF) for single-phase substances by sampling estimated fake RSFs from probability distributions parameterized by the RDF. This is promising, since it means we can extract more information about a material's microstructure (fake RSFs) from a measurement that is currently mostly used to determine macrostructure (RDF).

There are several steps I may try in the future to refine the estimated distribution of RSFs based on the RDF. One is to try sampling RSFs from a multivariate gaussian or dirichlet distribution parameterized by RDFs, since this would take into account the correlation between RSFs at different radii. Another is to learn how to generate RSFs from the RDF of a sample of mixed phase and unknown composition, then use the RSFs to estimate that unknown composition.

Contributions

I worked alone on this project, but was advised by Rodrigo Freitas from the Reed Materials Computation and Theory Group. He provided me with initial tools for data collection: the LAMMPS [4] script that I used to generate my raw data, as well as a script which then converted the raw data into RSFs. We met weekly to brainstorm next steps in the project that I would implement before the next meeting.

Project scripts and figures: https://github.com/16hchung/rsf_ml_2019

Citations

1. A. Sharp, Tristan & L. Thomas, Spencer & D. Cubuk, Ekin & Schoenholz, Samuel & Srolovitz, D.J. & J. Liu, Andrea. Machine learning determination of atomic dynamics at grain boundaries. Proceedings of the National Academy of Sciences. 115. 10.1073 (2018).
2. J. Behler, and M. Parrinello, PRL 98, 146401 (2007)
3. Molinero, Valeria, and Emily B. Moore. Water modeled as an intermediate element between carbon and silicon. The Journal of Physical Chemistry B 113.13 (2008): 4008-4016
4. S. Plimpton, Fast Parallel Algorithms for Short-Range Molecular Dynamics, J Comp Phys, 117, 1-19 (1995). http://lammps.sandia.gov
5. S. Schoenholz, et al., Nature Physics 12.5 (2016)