J Math Chem (2012) 50:2043–2050
DOI 10.1007/s10910-012-0019-5

ORIGINAL PAPER

Classification of nodal pockets in many-electron wave functions via machine learning

Erin LeDell · Prabhat · Dmitry Yu. Zubarev · Brian Austin · William A. Lester Jr.

Received: 20 March 2012 / Accepted: 6 April 2012 / Published online: 22 May 2012

Abstract Accurate treatment of electron correlation in quantum chemistry requires solving the many-electron problem. If the nodal surface of a many-electron wave function is available even in an approximate form, the fixed-node diffusion Monte Carlo (FNDMC) approach from the family of quantum Monte Carlo methods can be successfully used for this purpose. The issue of description and classification of nodal surfaces of fermionic wave functions becomes central for understanding the basic properties of many-electron wave functions and for the control of accuracy and computational efficiency of FNDMC computations. In this work, we approach the problem of automatic classification of nodal pockets of many-electron wave functions. We formulate this problem as that of binary classification and apply a number of techniques from the machine learning literature. We apply these techniques on a range of atoms of light elements and demonstrate varying degrees of success. We observe that classifiers with relatively simple geometry perform poorly on the classification task; methods based on a random collection of tree-based classifiers appear to perform best. We conclude with thoughts on computational challenges and complexity associated with applying these techniques to heavier atoms.

E. LeDell
Division of Biostatistics, UC Berkeley, Berkeley, CA 94720, USA

Prabhat
Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

D. Yu. Zubarev · W. A. Lester Jr. (✉)
Department of Chemistry, UC Berkeley, Berkeley, CA 94720, USA
e-mail: walester@lbl.gov

B. Austin
National Energy Research Scientific Computing Center, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

Keywords Binary classification · Machine learning · Many-body methods · Quantum chemistry · Fixed-node diffusion Monte Carlo · Electronic structure theory

1 Introduction

Two classes of methods account for the large majority of quantum chemical electronic structure calculations. Density functional theory (DFT) computes the molecular energy using a functional of the electron density [1,2]. Although DFT calculations are computationally inexpensive and often accurate, the true energy functional is not known, leaving DFT practitioners little recourse for systematically improving unsatisfactory results. Wave function-based theories (WFT), e.g. Hartree-Fock-Roothaan approach, [3] select a well defined, but approximate form of the electronic wave function and optimize the free parameters of that form. WFTs that account for electron correlation are typically significantly more expensive than DFT, but may converge to the exact wave function in certain limits. However, the accuracy of any practical application of WFT is limited by the constrained wave function form.

Quantum Monte Carlo (QMC) methods represent an important class of electronic structure simulation techniques used in quantum chemistry. QMC is rooted in stochastic approaches to solution of the electronic Schroedinger equation [4]. Compared to other techniques, QMC relies on the minimal number of approximations and can be used with wave functions ansätze of high complexity, which makes QMC highly accurate and flexible [5]. QMC calculations have great value for providing consistently reliable quantum chemical results and benchmarking and calibrating DFT, WFT and other methods. Diffusion Monte Carlo (DMC) uses stochastic projection in order to deliver eigenvalues of the molecular Hamiltonian by propagating the Schrödinger equation in imaginary time, $\boldsymbol{\tau} = it$

$$\frac{d}{d\boldsymbol{\tau}}\boldsymbol{\Phi}(\boldsymbol{X},\boldsymbol{\tau}) = (\underbrace{1/2\nabla^2}_{Diffusion} - \underbrace{(V(\boldsymbol{X}) - E_0)}_{Branching})\boldsymbol{\Phi}(\boldsymbol{X},\boldsymbol{\tau}) \tag{1}$$

where is $\boldsymbol{X}$ a position vector of an n-electron configuration and $\boldsymbol{\Phi}(\boldsymbol{X},\boldsymbol{\tau})$ is a wave function. This is accomplished by exploiting two isomorphisms. One is between the kinetic energy term in the Hamiltonian and a classical diffusion equation, the other is between the potential energy and a spatially inhomogeneous first-order rate equation. Diffusion can be simulated by a random walk and rate equation by a branching process. Treatment of fermionic systems requires a fixed-node constraint for DMC (FNDMC) to account for antisymmetry of the wave function and to avoid a collapse into the bosonic ground state [5]. In FNDMC, the domain of a many-electron wave function is separated into smaller "pockets" by the nodes of an approximate wave function so that DMC random walks are restricted to separate nodal pockets. The small energetic errors due to the FN approximation give FNDMC a level of accuracy commensurate with "gold-standard" methods of WFT [6]. The FN constraint constitutes the only uncontrolled approximation in FNDMC; any advances in characterizing and ultimately improving nodal pockets will have enormous theoretical impact on the field of electronic structure simulation. Understanding the geometry of these nodes

is a complex task, exacerbated by the fact that the dimensionality of the space under consideration grows proportionally with the number of electrons present in the system. An open challenge in the field is to develop effective analysis and visualization tools for better understanding the structure of these nodes.

The main result in the theory of nodal surfaces of many-electron wave functions is a so-called "tiling theorem" [7]. It states that nodal pockets of the Fermi ground state are the same within permutational symmetry. There is no such theorem for the excited states, which tremendously complicates treatment of such states. Visual inspection of Pfaffian wave functions of the simplest systems showed that there are tunnels between nodal pockets that are not present in the determinant wave functions [8]. Currently, plotting and visual inspection is the most typical approach to studies of nodal features [9,10]. Only limited understanding of the basic geometric properties of nodes of many-electron wave functions has been achieved to date, largely due to the complexity that arises with hypersurfaces in spaces of very high dimensionality. Development of a classification approach that creates a "representation" of a nodal structure can stimulate studies of nodal geometry and optimal properties of nodes. It will help to gain insights into the behavior of nodes of electronically excited states, which is currently not well understood.

The problem of classification of nodal structure of many-electron wave functions can be reformulated as the prediction of the sign of such a wave function for a given configuration of electrons. We seek to define a partitioning of the configuration space so that nodal pockets (i.e. regions of different signs) can be separated from each other. A potential approach towards solving this problem is to use data-driven mathematical constructs such as hyperplanes, trees, ensembles of trees, etc. A significant amount of classification machinery has been developed in the machine learning community over the past few decades, and the goal of this paper is to ascertain if we can leverage these techniques for the nodal pocket classification problem.

In this paper, we introduce a statistical approach to identification of nodal pockets. We present results from an investigation into the problem of automatically detecting nodal pockets of many-electron wave functions in electronic structure theory. We pose the task as that of binary classification (i.e., predicting the sign of the wave function) for high dimensional many-electron configurations resulting from QMC simulations. We then apply a number of modern classification techniques from the statistical literature and report on the observed performance, limitations and implications for future work.

2 Methods

Datasets for the study were generated using variational Monte Carlo (VMC) as implemented in Zori QMC software [11]. First row atoms from Li to Ne were considered. Trial wave functions in the product form were constructed from HF/STO-3G determinants and simple Jastrow correlation functions [12]. Simulations were performed with ensembles of 100 random walkers (or Markov chains) each propagated for 10,000 time-steps. Each electron is assigned a spatial x, y, z position, hence we have a 3n

dimensional vector representing an n-electron configuration. In our case, we had data ranging from Li (3 electrons, 9 dimensions) to Ne (10 electrons, 30 dimensions).

We used R [13] for performing all analysis and classification tasks reported in this paper. We used the classification and regression training (caret) package [14] which provides a convenient wrapper for tuning, training and testing various classification techniques. We used the following classification techniques: Support Vector Machine (SVM) (with linear and polynomial kernels), [15] Classification Trees, [16,17] K-nearest neighbor, [18] and Random Forest [19,20]. All of the tests were performed on a high-end Sun Microsystems Sunfire x4640 SMP machine, which consists of a single node with eight 6-core Opteron 2.6 GHz processors sharing 512 GB of memory. We note that in spite of the computational horsepower available, we were constrained to using single threaded implementations for individual classification techniques.

We split each dataset randomly into five sections with 200 K entries each. The first section was used exclusively for tuning purposes. The other four sections were used in a 4-fold cross validation fashion for determining final classification error for each technique. We now report on the results of our experiments.

3 Results and discussion

As indicated earlier, we used caret for tuning various classification techniques on an exclusive tuning dataset. We considered the following variables for the tuning process with respective designations in parenthesis: value of K (K) for K-nearest neighbor, tree max depth (Maxdepth) for classification trees, regularization constant indicating cost of constraint violation (Cost) for SVM with linear kernel, regularization constant indicating cost of constraint violation (Cost) for SVM with polynomial kernel, degree of polynomial (Degree) for SVM with polynomial kernel, and number of predictors sampled for splitting at each node (mtry) for Random Forest.

We could have optimized the Random Forest procedure for number of trees and depth of each tree. Due to runtime considerations (elaborated later in the paper), we empirically determined 500 trees and the default maxdepth value 30 to be reasonable. We plan on exploring this issue further in future work. Table 1 shows optimal settings learned from the tuning procedure for each method.

Our goal in tuning the five techniques for the first four elements (Li, Be, B, C) was to gain a sense for which techniques would perform best. For the remaining elements

Table 1 Optimal settings of tuning variables determined for each dataset by the tuning procedure

Technique, variable	Li	Be	B	C	N	O	F	Ne
SVM-linear, cost	0.25	1	0.25	2	–	–	–	–
SVM-polynomial, cost	0.25	0.75	0.25	0.25	–	–	–	–
SVM-polynomial, degree	2	4	4	3	–	–	–	–
Classification tree, maxdepth	7	12	12	9	–	–	–	–
K-nn, K	9	9	9	9	–	–	–	–
Random Forest, mtry	3	3	7	8	6	9	7	7

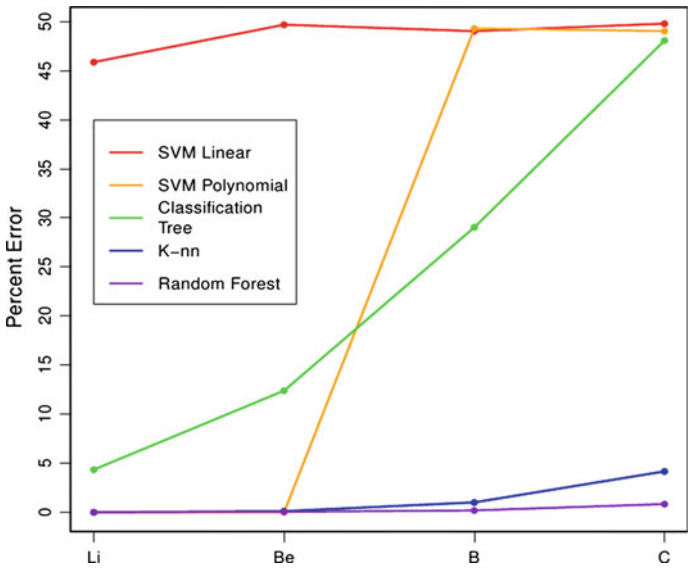

Fig. 1 Error rates for Li/Be/B/C across several classification techniques

(N, O, F, Ne) we chose to tune only the Random Forest technique because we felt it was the most promising for classifying performance.

After determining optimal parameters in the tuning phase, we did 4-fold cross-validation to test classification performance on the remaining sections of the dataset. We first report classification performance for Li, Be, B and C in Fig. 1. We make the following observations:

(a) Random Forest and K-nn appear to give the best classification performance among the methods that we considered. We get prediction error rates of 0.00 % (Li), 0.11 % (Be), 1.01 % (B), 4.17 % (C) using K-nn and 0.01 % (Li), 0.05 % (Be), 0.19 % (B), 0.84% (C) using Random Forest. This result is consistent with the general machine learning community literature in that Random Forests provide state-of-the-art performance for classification tasks.

(b) The classification task appears to get harder as we increase the number of electrons in the atom. This is to be expected since as we add electrons with each element, the dimensionality of the input space increases proportionally and the shape of the nodal pockets becomes more complex. An open question at this point is whether the Random Forest technique will work well for elements beyond C. We investigated this question further by testing the approach on elements N, O, F and Ne.

Figure 2 shows the time taken to run the Random Forest procedure on all datasets. This figure shows that the procedure is fairly computationally intensive; the task can take ∼12 h to complete a training/testing iteration on these datasets. This makes it challenging to tune the implementation for various parameters. Nevertheless, we performed rudimentary optimizations to the implementation for each of the elements reported in the paper.

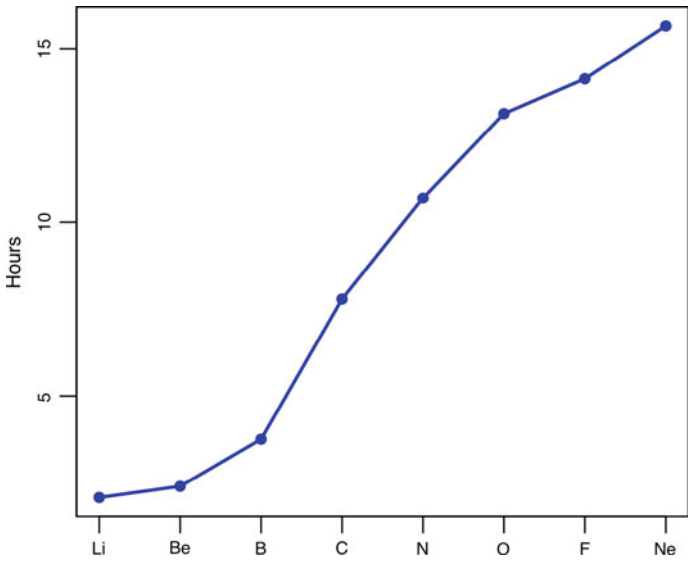

Fig. 2 Runtime performance of Random Forest method measured as time to train the Random Forest with 500 tuned trees

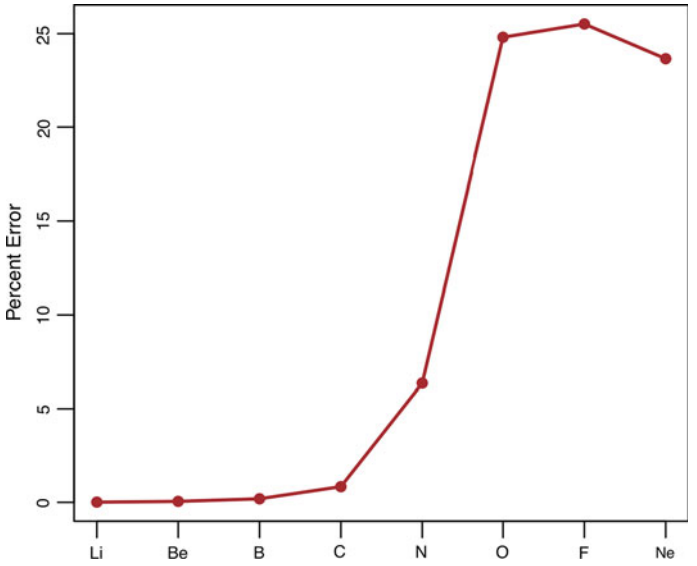

Fig. 3 Classification performance of Random Forest measured as percent of error of a Random Forest with 500 trees

Perhaps more interestingly, Fig. 3 shows that the performance of the Random Forest technique drops rather sharply as we examine elements with larger number of electrons. We observe an error rate of 6 % (N), 25 % (O), 25 % (F) and 24 % (Ne). This trend seems to imply that our current Random Forest implementation will have

limited success in classifying heavier atoms, and that we should either tune the implementation more carefully, or consider more comprehensive ensemble classification techniques such as Super Learner [21].

We have reported classification results for the first row of the periodic table (Li, Be, B, C, N, O, F, Ne). As the number of electrons in the atom increases across the row, the dimensionality of the configuration space increases and the spatial pattern of nodal pockets becomes more complicated. Most classification techniques used in our paper (SVMs with linear and polynomial kernels and classification trees) have relatively simple geometric interpretation but do not fare well.

While K-nn appears to be moderately competitive, we cannot use the technique from a science-application point of view since we do not end up with a geometric description of the nodal pockets at the end of the day. We will also need to keep the training data around in order to make predictions. This is infeasible for our use case.

The Random Forest method appears to do well initially, but its performance seems to degrade significantly for the latter elements considered in the study. We postulate a few reasons for this performance. We may need more data (i.e. more walkers) to populate a larger configuration space. We might need to do more extensive tuning for the heavier elements. The shape of nodal pockets is simply too complex to be represented well by a Random Forest. While it is easy to generate a larger amount of data with our simulation code, with the current implementation of Random Forest, it is unclear if we can run the procedure in a reasonable amount of time. In retrospect, it would have been convenient to have access to a parallel implementation of the Random Forest package. While we can manually carry out cross-validation and tuning in parallel, we are still limited by the time to train the Random Forest procedure on a single dataset. It produced good classification for the elements Li, Be, B and C. The performance for N, O, F and Ne was not as stellar; we would like to tune the method on larger datasets. It is possible that other techniques such as gradient boosting [22] might provide a further improvement in performance; we will investigate this in the future. We would like to integrate our prediction framework into the Zori code and use the predicted sign value to provide an alternative method for enforcing the FN constraint. This step is perceived as a route to direct optimization of nodal surfaces that is not limited by the flexibility of a particular trial wave function. We would also like to assess to what degree the presence of misclassified points is detrimental considering the stability of QMC simulations that rely on the FN approximation. An important but challenging problem for future work is to develop an approach for comparative analysis of the learned Random Forests and to gain insights into the corresponding nodal surfaces. This task is key to rationalization of changes in nodal structure upon transitions among different electronic states of the same molecular system.

4 Conclusions

In this paper, we have approached an open problem in QMC: can we automatically classify nodal pockets in the electronic structure of molecules. We have applied a number of modern classification techniques from the statistical literature with varying degrees of success. We found that Random Forest outperforms other methods across

the board for early elements in the periodic table. However, as we applied Random Forest to elements with larger numbers of electrons, we were unable to achieve good performance. We postulate that this might be due to lack of sufficiently representative data, or simply a reflection of the complexity of nodal pocket geometry. We believe that further development of powerful ensemble-based classification procedures, and access to computationally efficient, parallel algorithmic implementations will be important for tackling this problem in the future.

Acknowledgments This work is supported by the Director, Office of Laboratory Policy and Infrastructure Management of the U.S. Department of Energy under Contract No. AC02-05CH11231. D.Y.Z. was supported by the National Science Foundation under Grant NSF CHE-0809969. W.A.L. was supported by the Director, Office of Energy Research, Office of Basic Energy Sciences, Chemical Sciences, Geosciences and Biosciences Division of the U.S. Department of Energy, under Contract No. DE-AC03-76F00098. This research used computational resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the US Department of Energy under Contract No. DE-AC02-05CH11231.

References

1. P. Hohenberg, W. Kohn, Phys. Rev. B **136**, B864 (1964)
2. W. Kohn, L.J. Sham, Phys. Rev. B **140**, 1133 (1965)
3. C.C.J. Roothaan, Rev. Mod. Phys. **23**, 69 (1951)
4. B.L. Hammond, W.A. Lester Jr., P.J. Reynolds, *Monte Carlo Methods in Ab Initio Quantum Chemistry: Quantum Monte Carlo for Molecules* (World Scientific, Hackensack, 1994)
5. B.M. Austin, D.Yu. Zubarev, W.A. Lester Jr., Chem. Rev. **112**, 263 (2012)
6. J.C. Grossman, J. Chem. Phys. **117**, 1434 (2002)
7. D.M. Ceperley, J. Stat. Phys. **63**, 1237 (1991)
8. M. Bajdich, L.K. Wagner, G. Drobny, L. Mitas, K.E. Schmidt, Phys. Rev. Lett. **96**, 130201 (2006)
9. W.A. Glauser, W.R. Brown, W.A. Lester, D. Bressanini, B.L. Hammond, M.L. Koszykowski, J. Chem. Phys. **97**, 9200 (1992)
10. D. Bressanini, P.J. Reynolds, Phys. Rev. Lett. **95**, 110201 (2005)
11. A. Aspuru-Guzik, R. Salomon-Ferrer, B. Austin, R. Perusquia-Flores, M.A. Griffin, R.A. Oliva, D. Skinner, D. Domin, W.A. Lester Jr., J. Comput. Chem. **26**, 856 (2005)
12. R. Jastrow, Phys. Rev. **98**, 1479 (1955)
13. The R Project for Statistical Computing, http://www.r-project.org/
14. R caret package, http://cran.r-project.org/web/packages/caret/index.html
15. C. Cortes, V. Vapnik, Mach. Learn. **20**, 237 (1995)
16. L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, *Classification and Regression Trees* (Wadsworth & Brooks/Cole Advanced Books, Monterey, 1984)
17. T. Hastie, R. Tibshirani, J.H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer, New York, 2001)
18. T.M. Cover, P.E. Hart, IEEE Trans. Inform. Theory **13**, 21 (1967)
19. L. Breiman, Mach. Learn. **45**, 1 (2001)
20. T. Ho, IEEE Trans. Patt. Anal. Mach. Intell. **20**, 832 (1998)
21. R SuperLearner package, http://cran.r-project.org/web/packages/SuperLearner/index.html
22. J.H. Friedman, *Stochastic Gradient Boosting* (Stanford University, Stanford, 1999)