Estimating Convergence Probability for the Hartree-Fock Algorithm for Calculating Electronic Wavefunctions for Molecules

Sofia Izmailov, Fang Liu

December 14, 2012

1 Introduction

In quantum mechanics, molecules can be described by their wavefunctions. The wavefunction is given by the Schrodinger equation:

$$H\Phi = E\Phi$$

where H is the Hamiltonian operator and E is the energy of the system described by Φ. The Schrodinger equation cannot be solved exactly for molecules larger than a hydrogen atom. Therefore, certain assumptions are often made to simplify the problem. The Born-Oppenheimer approximation is frequently used and it treats nuclei as stationary because their velocity is much lower than that of the electrons. This allows the wavefunction to be broken into the nuclear and electronic components. For many applications in chemistry we are concerned with solving the electronic wavefunction. Since the Schrodinger equation cannot be solved exactly, basis sets (e.g. Gaussian-type functions) are used to approximate electronic wavefunctions.

The Hartree-Fock algorithm calculates the electronic wavefunction for a system given some basis set. The algorithm tries to solve a system of non-linear equations using an iterative procedure. Specifically, the problem requires a solution to a eigenvector equation in the form $\mathbf{FC} = \mathbf{C}\epsilon$ where the Fock matrix $\mathbf{F}$ is dependent on $\mathbf{C}$. Convergence is reached when certain properties of the system stabilize (e.g. Energy reaches a stable minimum or the basis set coefficients stabilize). However, sometimes the algorithm converges very slowly, oscillates around some value, or even diverges. Many modifications to the original algorithm exist to increase convergence rate and decrease the likelihood of failure.

One such algorithm is the directed inversion in the iterative subspace (DIIS) method [1]. In this algorithm, we determine an error vector for each iteration associated with the Fock matrix used in that iteration. The error for the next step is approximated as a linear combination of some n previous error vectors $e_{n+1} = \sum_i^n c_i(e_i)$ with the constraint that $\sum_i^n c_i = 1$. The goal is to minimize this approximation of the error so this problem can be solved by standard the standard Lagrange multiplier method. The $\mathbf{c}$ coefficients are than used to update the Fock matrix for the next step. Then the actual e_{n+1} error may be calculated from the $\mathbf{F}_{n+1}$. The process is repeated until e_n is sufficiently small.

While the DIIS method does increase convergence rate and decrease failure rate in electronic structure calculations, some systems continue to fail to converge or converge slowly. Since the calculations of each step of the Hartree-Fock algorithm can be really expensive, especially when large molecules or large basis sets are used, it would be very useful to have a method of determining whether a current calculation is going to converge soon. This way, one may choose whether to continue on with an existing calculation or to start over and try a different set of initial conditions which may lead to a more successful calculation.

2 Generating Data

Hatree-Fock calculation with DIIS algorithm was carried out with TeraChem[2] quantum chemistry package. We chose benzene as the model molecule to generate examples of Hatree-Fock calculation data, and the detailed process is described as follows.

80 different conformations of benzene were generated with AMBER[3] molecular dynamics package. Then the Hartree-Fock calculation for these conformations were conducted with Terachem. The 6-31G basis set was employed throughout the calculation to make sure that the size of error matrices remained the same for all calculations. Maximum number of DIIS iterations was set to 100, so the calculation was terminated if not converged within 100 steps. Finally, 80 different examples were obtained, with 20 easily converged ones (within 13 iterations), 20 slowly converged ones (26-88 iterations), and 40 non-converged ones. For each example, the calculation result was composed of history of total energy difference between two consecutive iterations $\Delta \mathrm{E}_i$, DIIS error matrix for each iteration $\mathbf{e}_i$, and the DIIS linear equation matrix $\mathbf{B}$, which was generated for minimizing the approximated error $e_{n+1} = \sum_i^n c_i(e_i)$ with lagrange multiplier method:

$$\begin{pmatrix} 0 & -1 & -1 & \cdots & -1 \\ -1 & B_{11} & B_{12} & \cdots & \\ \vdots & & & & \\ -1 & \cdots & & B_{ij} & \\ \vdots & & & & \end{pmatrix} \begin{pmatrix} -\lambda \\ c_1 \\ \vdots \\ c_i \\ \vdots \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \\ \\ 0 \\ \vdots \end{pmatrix}$$

where $\mathbf{B}_{ij}$ the scalar product of any two error matrices for the last 10 iterations: $\mathbf{B}_{ij} = Tr(e_i, e_j^+)$. $\Delta \mathrm{E}_i$ and $\mathbf{B}$ were chosen as input features.

A feature vector was generated for every step in each iteration. For the response variables, steps which were more than n steps from convergence were labeled -1 and steps that were n or fewer steps from convergence were labeled as $+1$. We used the values $n = 3, 5, 10$ to create three sets of response variables.

3 Modeling the Data

We used the LIBSVM package[4] to generate severals SVM models for our data. The quality of the models was assessed by random 5-fold cross-validation of the training set data.

We had a training set of 5,153 trials. The features were: the number of molecular orbitals used in the model, the ΔE in calculated energy for the previous m steps, the $\mathbf{c}$

vectors and λ values for the previous m steps, and the $\mathbf{B}_{ij}$ values for the previous $2m$ steps where m ranged from 1 to 10. Since many of the trials represented steps that were too early in the calculation to have m or $2m$ previous steps, we treated the calculation as if, prior to the actual first step, the calculation had simply failed to make any progress. That is, we set $\Delta E = 0$ and we set the $\mathbf{c}$, λ, and $\mathbf{e}$ (which is used to calculate $\mathbf{B}_{ij}$) to the values they have at the first step in the energy calculation.

Number of previous steps used to form features (m)	Number of features f_m
1	16
2	35
3	58
4	85
5	116
6	151
7	190
8	233
9	280
10	331

We used the C-SVC with a radial kernel $exp(-\frac{1}{f_m}|u-v|^2)$ where f_m is the number of features generated from taking the previous m steps into account. In order to evaluate the quality of our models, we looked at the overall accuracy of prediction, the f-score, and the area under receiver operating characteristic curve. We looked at how feature size affected these metric for the three sets of response variables we created for different convergence distances. For any given feature set and metric we optimized the cost C to give the highest value of the metric possible.

4 Results

Figures 1, 2, and 3 show the optimal accuracy, f-score, and area under ROC curve for feature sets $m = 1, \dots 10$.

The overall accuracy increases with feature set size. Accuracy is also higher when fewer data points are labeled as near convergence. It is to be expected that steps farther away from convergence would be more difficult to identify as near convergence. Accuracy for predicting convergence within 3 steps is very high but this could be due to overfitting. The same amount of data was used for all the SVM models but limiting the allowed distance from convergence also shrinks the data in the "near convergence category."

The area under the ROC curve for different convergence distances and feature set sizes shows two opposite trends. Predicting convergence in 10 steps has a larger area under curve for larger feature sets but predicting convergence in 3 or 5 steps has a smaller area under curve for larger feature sets. This decrease may also be due to the relative sizes of the two categories: "near convergence" and "not near convergence." The relative rate of false positive and true positives is changing there are few true positives to begin with. Increasing training set size would likely help make an upward trend like in the case of estimating convergence in 10 steps.

F-score increases slightly for predicting convergence in 10 steps and stays approximately constant or increases very slightly for the other trials. This may mean precision and recall are not changing much.

5 Future Work

The application of SVMs to determine whether a DIIS calculation is likely to converge soon has been promising. However, there is further work necessary to improve the quality and ensure the applicability to more generalized problems. A larger number of training samples would improve prediction quality. Additionally, it would be necessary to consider multiple molecules in order to be able to build an SVM which would be more generally applicable to systems other than benzene. With a large number of different molecules represented, SVMs can be generated for other basis sets as well.

References

[1] Pulay, P. (1982), Improved SCF convergence acceleration. J. Comput. Chem., 3: 556-560.

[2] I.S. Ufimtsev and T.J. Martinez, Quantum Chemistry on Graphical Processing Units. 3. Analytical Energy Gradients and First Principles Molecular Dynamics, J. Chem. Theory Comput., 2009, 5, p2619.

[3] D.A. Case, T.A. Darden, T.E. Cheatham, III, C.L. Simmerling, J. Wang, R.E. Duke, R. Luo, R.C. Walker, W. Zhang, K.M. Merz, B. Roberts, S. Hayik, A. Roitberg, G. Seabra, J. Swails, A.W. Goetz, I. Kolossvry, K.F. Wong, F. Paesani, J. Vanicek, R.M. Wolf, J. Liu, X. Wu, S.R. Brozell, T. Steinbrecher, H. Gohlke, Q. Cai, X. Ye, J. Wang, M.-J. Hsieh, G. Cui, D.R. Roe, D.H. Mathews, M.G. Seetin, R. Salomon-Ferrer, C. Sagui, V. Babin, T. Luchko, S. Gusarov, A. Kovalenko, and P.A. Kollman (2012), AMBER 12, University of California, San Francisco.

[4] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1-27:27, 2011. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm

Figure 1: Best Accuracy for convergence for different m values

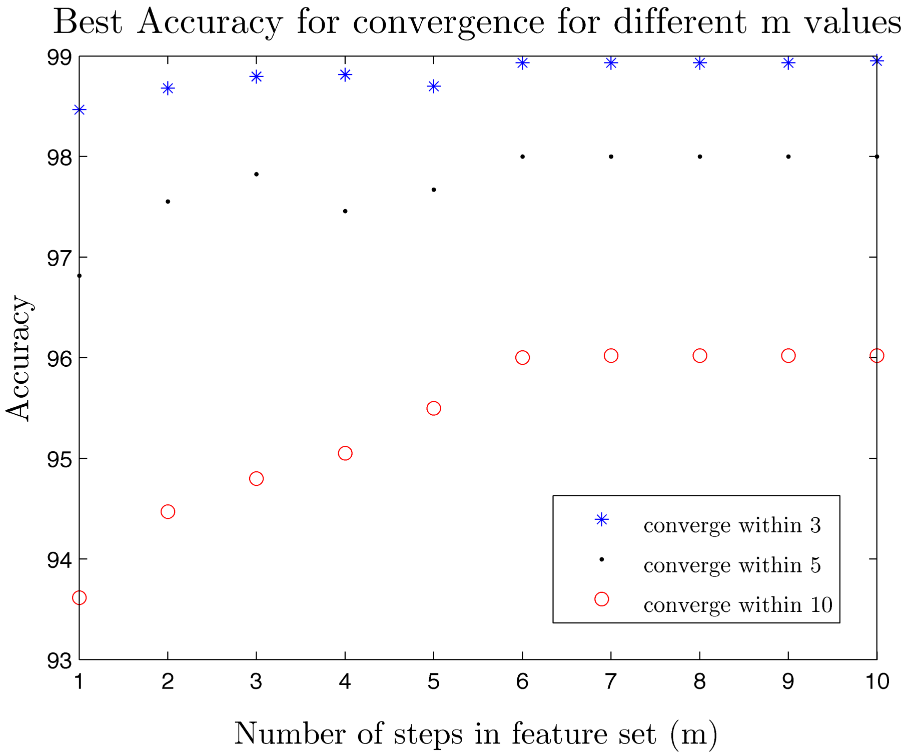


Figure 2: Best F-score for convergence for different m values

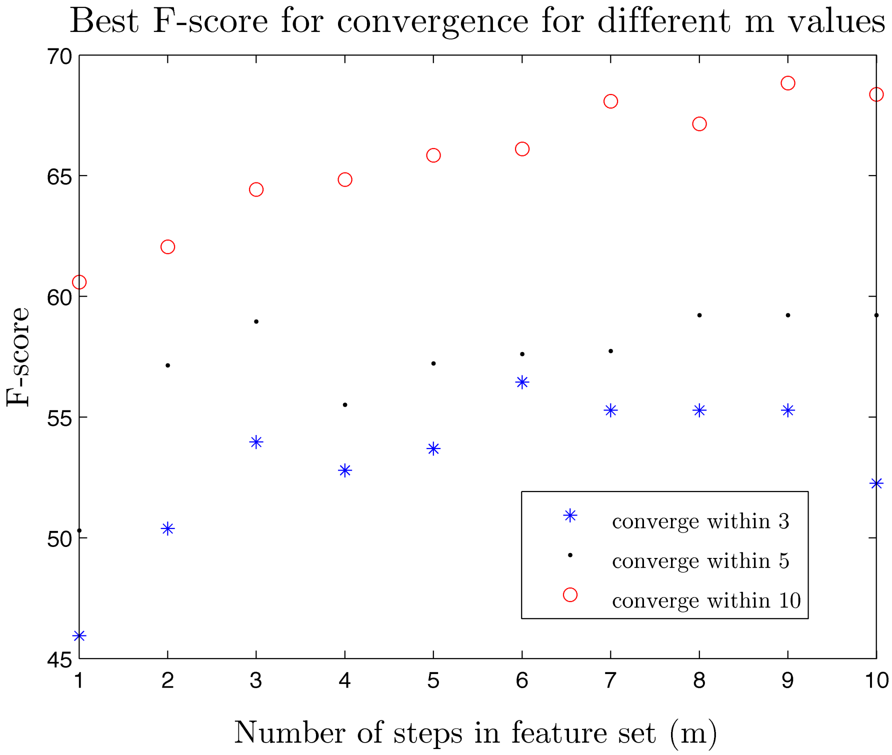
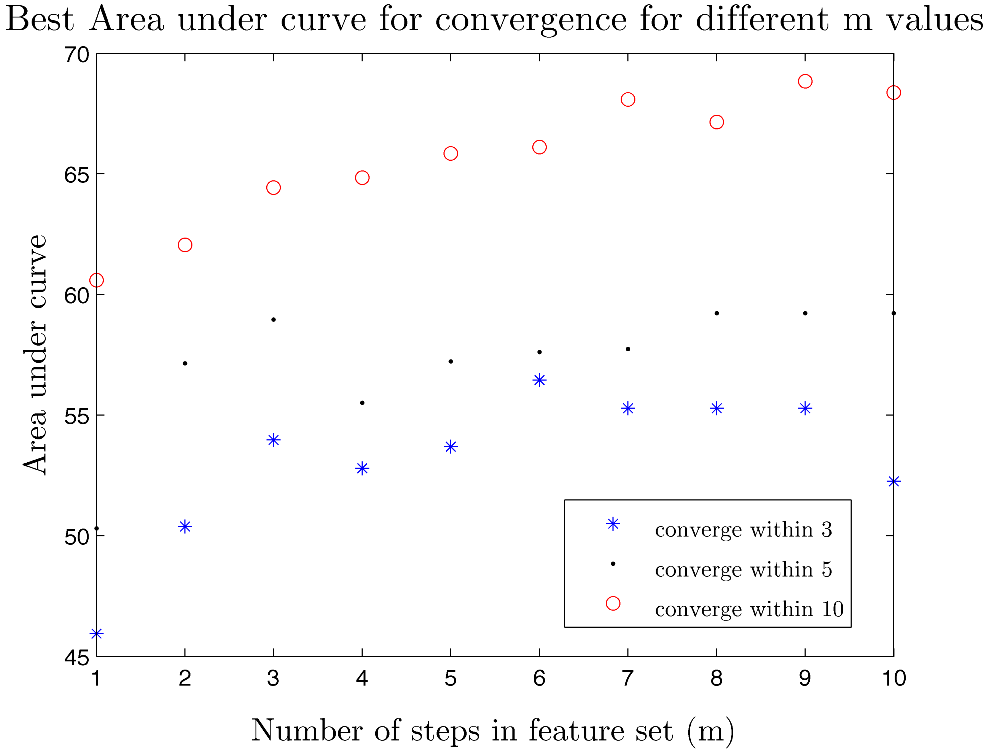


Figure 3: Best area under ROC curve for convergence for different m values


Best Area under curve for convergence for different m values
Area under curve
Number of steps in feature set (m)
✳ converge within 3
• converge within 5
○ converge within 10