

Project on Machine Learning

Data Analysis and Machine Learning FYS-MAT3155/FYS4155

Department of Physics, University of Oslo, Norway

Fall semester 2017

Using results from Monte Carlo models for machine learning

Introduction. The aim of this project is to use an already developed Monte Carlo program (either the Ising Model or a variational Monte Carlo code) to produce, in case of the [Ising model](#), the energy as function of temperature. For the [variational Monte Carlo calculation](#) of interacting electrons in an oscillator trap, the data are represented by the ground state energies as functions of the variational parameters.

In its simplest form the energy of the Ising model is expressed as, without an externally applied magnetic field,

$$E = -J \sum_{\langle kl \rangle}^N s_k s_l$$

with $s_k = \pm 1$. The quantity N represents the total number of spins and J is a coupling constant expressing the strength of the interaction between neighboring spins. The symbol $\langle kl \rangle$ indicates that we sum over nearest neighbors only. We will assume that we have a ferromagnetic ordering, viz $J > 0$. We will use periodic boundary conditions and the Metropolis algorithm only. Alternatively we can use the supplied variational Monte Carlo program which solves Schroedinger's equation for two interacting electrons in a harmonic oscillator trap. Both codes can be found at the webpage of the [course under programs](#).

Part a): Producing the data. If we opt for the Ising model code, we need to generate for every span over the lattice and output of the total energy and the magnetic moment (if we wish to study this quantity as well), that is compute and write to file as function of temperature $\langle E \rangle$ and $\langle |M| \rangle$. We will use a fixed lattice size of $L \times L = 40 \times 40$. Make sure the calculations have been equilibrated and compute these expectation values for $T \in [1.0, 3.0]$ with a step in temperature $\Delta T = 0.1$ or smaller. This are the data you will use to

estimate the standard deviation in the next part of the project. You should keep a file for the different temperatures as you will need to compute the standard deviation for each temperature.

Alternatively you can run the variational Monte Carlo program for two interacting electrons confined to move in a harmonic oscillator trap. Here you will produce the variational expectation value of the energy as a function of the variational parameters α and β . You could single out only a set of such parameters after you have found the minimum energy.

These two cases form then our training data which we will use for our estimates of the standard deviation.

Part b): Estimating the standard deviation. This part deals with widely used resampling methods to find the standard deviation on our data sets. The methods we can use are the Blocking method, the Bootstrap method and the Jackknife method. The latter two are examples of so-called resampling methods. [The functions provided under the program folder](#) encode all three methods. The typical situation is to generate data using a C++ or Fortran code, as in part a) above. These data are then used by a post-analysis program in order to perform a statistical analysis.

In case of the Ising model, the final product will be a table of the expectation values of the energy (or the magnetic moment) as function of temperature and with a proper standard deviation. These data will then enter our analysis in parts c-e). We will call these data the training data.

If you opt for the variational Monte Carlo program, you need to make a table of the ground state energy as function of various variational parameters (you can limit yourself to α only) with proper standard deviation estimations.

Your task here is thus to use the data from part a) and generate the standard deviation using one (or more) of the methods discussed above. In general, the Bootstrap method is the most widely used one.

These data will then be used in our next step, where we will try to obtain a functional approximation to our data sets.

Part c): Fitting the data using regression analysis and other methods. More text to come

Part d): Introducing Bayesian statistics. More text to come

Part e): Studying the Ising model or the VMC results with Neural networks. More text to come

Background literature

If you wish to read more about the Ising model and statistical physics here are three suggestions.

- M. Plischke and B. Bergersen, *Equilibrium Statistical Physics*, World Scientific, see chapters 5 and 6.
- D. P. Landau and K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics*, Cambridge, see chapters 2,3 and 4.
- M. E. J. Newman and T. Barkema, *Monte Carlo Methods in Statistical Physics*, Oxford, see chapters 3 and 4.

Introduction to numerical projects

Here follows a brief recipe and recommendation on how to write a report for each project.

- Give a short description of the nature of the problem and the eventual numerical methods you have used.
- Describe the algorithm you have used and/or developed. Here you may find it convenient to use pseudocoding. In many cases you can describe the algorithm in the program itself.
- Include the source code of your program. Comment your program properly.
- If possible, try to find analytic solutions, or known limits in order to test your program when developing the code.
- Include your results either in figure form or in a table. Remember to label your results. All tables and figures should have relevant captions and labels on the axes.
- Try to evaluate the reliability and numerical stability/precision of your results. If possible, include a qualitative and/or quantitative discussion of the numerical stability, eventual loss of precision etc.
- Try to give an interpretation of you results in your answers to the problems.
- Critique: if possible include your comments and reflections about the exercise, whether you felt you learnt something, ideas for improvements and other thoughts you've made when solving the exercise. We wish to keep this course at the interactive level and your comments can help us improve it.
- Try to establish a practice where you log your work at the computerlab. You may find such a logbook very handy at later stages in your work, especially when you don't properly remember what a previous test version of your program did. Here you could also record the time spent on solving the exercise, various algorithms you may have tested or other topics which you feel worthy of mentioning.