Finding Optimal Minimum Energy Pathways using Graphical Models

Applying the max-product algorithm to find the minimum energy pathway and avoid local minima.

Adam Carruthers

Abstract

We explore the issue of local minima when finding minimum energy pathways in energy landscapes. We then apply the max-product algorithm to efficiently explore the landscape and hence find a better route. Finally we evaluate the performance of this approach in different scenarios and with respect to different parameters.

Chapter 1

Introduction

Energy landscapes can describe an incredible array of systems, particularly biological and chemical processes. It is therefore vital to understand the stable states in this system, and the transitions between them. For example, if you can find the transition pathway between two stable states in a chemical reaction that has the lowest possible maximum energy then it is possible to deduce information about reaction speed. Such a pathway is called the minimum energy pathway.

A significant amount of research has been devoted to finding these minimum energy pathways. A popular method is called Nudged Elastic Band (NEB), which lets the path be a string of points. It starts with a random guess (normally a straight line between two stable points), and then lets the points fall down the slope of the energy landscape (given by the gradient of the energy function). However it then forces the points to remain equidistant by exerting a spring force on them[1].

This algorithm is a form of gradient descent. As with any form of gradient descent the algorithm can get stuck in local minima, as in Figure 1. When the gradient descent is run the path gets stuck in a high energy valley, when going around gives a lower energy result.

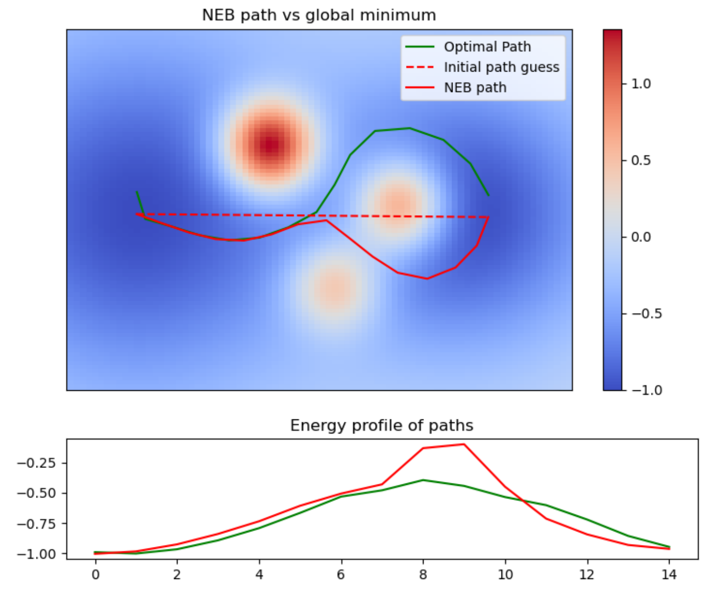

Figure 1.1: The NEB generated and global minimum energy path between two minima (above) with the energy profile of those paths (below).

[1]This is a slight simplification, as moderations are made to stop the algorithm cutting corners. See the paper cited for implementation details.

In this paper we propose an algorithm that can efficiently explore the landscape and the possible paths. This algorithm is therefore able to find very promising initial path guesses that outperform naive NEB, and it even has the ability to quickly find a selection of paths that go through different areas of the energy landscape. We provide an analysis of how this algorithm performs in different situations - including the time it takes to run as well as the quality of the outputted paths. We will refer to this algorithm as the Markov Random Field (MRF) algorithm, as the way we build up the paths uses techniques developed for Markov Random Fields. The main chunk of the algorithm is actually a technique called Max-Sum Message Passing, but I'll refer to the wider technique as the MRF to encapsulate the full process.

1.1 Basic MRF algorithm description

The algorithm effectively defines the path by a number of points. It works along the path step by step, testing a number of positions for each point and working out the how good that possible position is. When working out the quality of a position it incorporates the quality of all the points leading up to that position using an algorithm called max-produce message passing. If effectively scans the points that lead up to each position and picks the preceeding path that maximises its quality.

Once we have found the path we then smooth out the generated path with the Nudged Elastic Band method, as if our suggested path is close to the minimum path then NEB should converge to it. This is important since the algorithm must perform a very coarse scan of the landscape for time complexity reasons. The algorithm scales linearly by the number of positions being scanned, multiplied by the number of positions on average that one position is connected to. This means that particularly in high dimensions, with a lot of space that you need to explore, the produced path cannot be very fine grained.

This method leaves us with a few important questions, which we will consider in the paper:

- How do we decide the possible positions and transitions that our path could go through?
- How do we define the quality of a point or a transition?
- How do we make it all fit together?
- How long does this algorithm take to run? How does this compare to NEB?
- Are the outputs of this algorithm better than naive NEB?

Chapter 2

Algorithm structure

Some text...