Mini-Project #3

Due by 11:59 PM on Tuesday, April 21st.

Instructions

You can work individually or with one partner. If you work in a pair, both partners will receive the same grade. Detailed submission instructions can be found on the course website (`http://cs168.stanford.edu`) under "Coursework - Assignment" section. If you work in pairs, **only one member** should submit all of the relevant files.

If you've written code to solve a certain part of a problem, or if the part explicitly asks you to implement an algorithm, you must also include your code in your solution to that part **in your pdf submission**.

Reminder: No late assignments will be accepted, but we will drop your lowest assignment grade when calculating your final grade.

Background

How long does it take to visit 30 national parks?[1] The TAs plan on having an epic road trip this summer, and have picked 30 of the 59 national parks to visit. A proper route starts and ends at the same national park, and visits each park exactly once. Your goal in this project is to help them find the shortest route possible.

In the file `parks.csv`, you'll find the longitude and latitude of the 30 national parks *in alphabetical order of the park names*. To simplify the problem, we assume that the distance between two national parks is as follows:[2]

$$\text{distance} = \sqrt{(\text{longitude}_1 - \text{longitude}_2)^2 + (\text{latitude}_1 - \text{latitude}_2)^2}$$

As a sanity check, the distance between Acadia to Arches (the first two in the file) is 41.75 units, and the length of the route that visits all parks in alphabetical order (Acadia $\rightarrow$ Arches $\rightarrow$ Badlands $\rightarrow \cdots \rightarrow$ Acadia) is 491.92 units.

Part 1: Markov Chain Monte Carlo (MCMC)

Goal: To gain experience exploring a vast search space using Markov Chain Monte Carlo (MCMC). Also, to explore the cost-benefit trade-off as one interpolates between local search and random search via a "temperature" parameter.

Description: In this part, you will implement MCMC to explore the space of traveling salesman tours of the national parks. The MCMC search will be parameterized by a "temperature" parameter, T, which governs the trade-off between local search and random search. To begin with, consider the MCMC algorithm that starts with a random permutation of the N parks; at each iteration, randomly select two successive parks, and propose the route defined by switching their order. If the proposed route has a lower total distance than the current route, set the current route to be the proposed route; otherwise, if $T > 0$, with probability $e^{-\Delta_d/T}$, update the current route to be the proposed route, where Δ_d is the increase in total distance.

[1] This problem is a version of the Traveling Salesman Problem (TSP): `http://en.wikipedia.org/wiki/Travelling_salesman_problem`.

[2] The proper measure (without considerations for traffic, elevation, or the existence of roads) is Vicenty's formulae (`http://en.wikipedia.org/wiki/Vincenty's_formulae`), but we'll spare you the trouble.

Repeat this process MAXITER times. The algorithm outputs the shortest route found during the MAXITER iterations (which need not be the last route). The following is pseudo-code for the above algorithm:

```
route ← random permutation of the N parks
best ← route
for i = 1, 2, . . . , MAXITER do
    route_new ← route with the order two random successive parks reversed
    Δ_d = distance(route_new) - distance(route)
    if Δ_d < 0 OR (T > 0 AND random() < e^(−Δ_d/T)) then
        route ← route_new
    end if
    if distance(route) < distance(best) then
        best ← route
    end if
end for
```

(a) Implement the MCMC algorithm. How many states are in this Markov chain? If MAXITER tends to infinity, will you eventually see all possible routes? [Hint: consider separately the cases of $T = 0$ and $T > 0$.]

(b) Set MAXITER to be $10,000$. Run the algorithm in four different regimes, with $T \in \{0, 1, 10, 100\}$. For each value of T, run 10 trials. Produce one figure for each value of T. In each figure, plot a line for each trial: the iteration number against the length of the current value of route during that iteration. (Your solution to this part should have 4 figures, and each plot should contain 10 lines.)

(c) Discuss the costs and benefits of using a higher/lower temperature. Feel free to refer to specific features of the figures you produced in the previous part. Among the 4 values of T, which one seems to work the best? [Hint: some keywords you can use for discussion are mixing time, local optima, and global optima.]

(d) Modify the above MCMC algorithm as follows: in each iteration select two parks at random (*not necessarily successive parks*) and propose the route obtained by switching them. Repeat the experiments from part (b). Describe whatever differences you see relative to part (b), and propose one or more explanations for why this Markov Chain performs differently from the previous one.

Deliverables: Code and answers for part (a). Figures for part (b). Discussion for part (c). Figures and discussion for part (d).

Part 2: Simulated Annealing

Goal: To gain experience with Simulated Annealing, a technique for improving over MCMC by varying the temperature with the number of iterations, rather than using a fixed temperature. Also, to explore whether or not Simulated Annealing is a win in our running TSP example.

Description: Simulated Annealing (SA) is an approach that attempts to combine the merits of both low and high temperatures in MCMC. It mimics the physical process of heating up a material and slowly cooling it down to find its lowest energy state. In our setting, it corresponds to decreasing the value of T as the algorithm proceeds. In this part, you will investigate a square-root annealing schedule:

$$T = \frac{c}{\sqrt{t}}$$

where t is the current iteration number, and c is a constant (defined below).

(a) Modify your code from Part 1(d) to incorporate Simulated Annealing. (Make sure you save a copy of the original code somewhere.) Set MAXITER to be $10,000$ and c to be 70. Run 10 trials. Produce a figure analogous to the ones in 1(d).

(b) Compare the figure from the previous part to those you produced in 1(d). Does Simulated Annealing seem to improve the results you get? (Be honest!)

(c) Plot the best route that you got (across all 10 trials). In your plot, use the x coordinate for longitude and the y coordinate for latitude; draw a line between each pair of successive parks in the route. Does the result look close to an optimal solution? Describe at least one way to improve the algorithm.

(d) [Optional] Inspired by your answer from the previous part, the TAs now want to tour all 59 national parks and are running a contest to find the shortest route. You can find the complete list of national parks in the `parksContest.csv`.[3] To enter the contest, submit a permutation of 1..59 and the length of the route derived from that permutation; additionally, describe in a sentence or two how you obtained this permutation. No points are assigned to this part, but the team that submits the shortest route may receive a special prize.

Deliverables: Code and figure for part (a). Discussion for part (b). Plot and discussion for part (c). Permutation and length for part (d).

[3] You can also find them here: `http://en.wikipedia.org/wiki/List_of_national_parks_of_the_United_States`.