ESE 650 Homework 5 - Reinforcement Learning

Tarik Tosun

April 8, 2014

Introduction

In this assignment, I implemented reinforcement learning for path planning. The program learns to plan appropriate paths for different transportation modalities (driving and walking) by imitating paths chosen by a human. Path planning is done using Dijkstra's algorithm and a cost map, where the cost is a linear function of a feature vector extracted from each pixel of the map (a JPEG image of Penn's campus). I used a method based on the LEARCH algorithm presented in [1] and in class to learn optimal cost function parameters by gradient descent.

Algorithm and Feature Set

Cost Function and Training

I used the loss function and cost function presented in class:

$$J = \sum_t C\left(\vec{x}_{des}(t)\right) - \sum_t C\left(\vec{x}_{opt}(t)\right)$$

$$C\left(\vec{x}(t)\right) = \exp\left(\sum_i w_i h_i\left(\vec{x}(t)\right)\right)$$

The cost function is the exponential of a linear function of the feature vector at a pixel, and the loss function is the difference between the cumulative costs of the desired and optimal path between two points. We find $\vec{w}^* = \arg\min_{\vec{w}} J$ through an iterative gradient descent procedure:

$$\left(\nabla_{\vec{w}} J\right)_i = \sum_t C\left(\vec{x}_{des}(t)\right) h_i\left(\vec{x}_{opt}(t)\right) - \sum_t C\left(\vec{x}_{opt}(t)\right) h_i\left(\vec{x}_{opt}(t)\right)$$

$$\vec{w}_{new} \leftarrow \vec{w}_{old} + \eta(\vec{\nabla_w J})$$

I found that my planner was able to generalize to new paths without using cost augmentation, so I did not use it in the results presented here.

Figure 1: Map labeled using K-Means, K=16. Each color in the image corresponds to a different label.

Features

To generate a feature vector at each pixel in the image, I ran K-Means on image with $K = 16$, establishing a 16-color "palette" to describe a given pixel. Rather than labeling each pixel with a single cluster, I used the inverse exponential of the distance from the pixel to each cluster center as features. My thinking was this continuous feature space would have more descriptive power than a discrete set of labels, and would allow my linear classifier to do the discrimination rather than K-Means.

Looking at the map, I noticed that most of the semantically important regions of the map (i.e.: regions corresponding to roads, buildings, trees, etc) consisted of a patchwork of very differently colored pixels, but on average had a fairly consistent color. In order to reduce the high-frequency content of the image, I applied a gaussian blur before performing K-Means. Figure 1 shows the map labeled by cluster.

Results

Figures 2 and 3 show the cost maps from training on driving and walking modalities. Figures 4 and 5 show some example paths planned for driving and walking. To try some paths for yourself, execute the script `RUN_ME.m` in my submission, which will allow you to (quickly and easily) plan paths on the map.

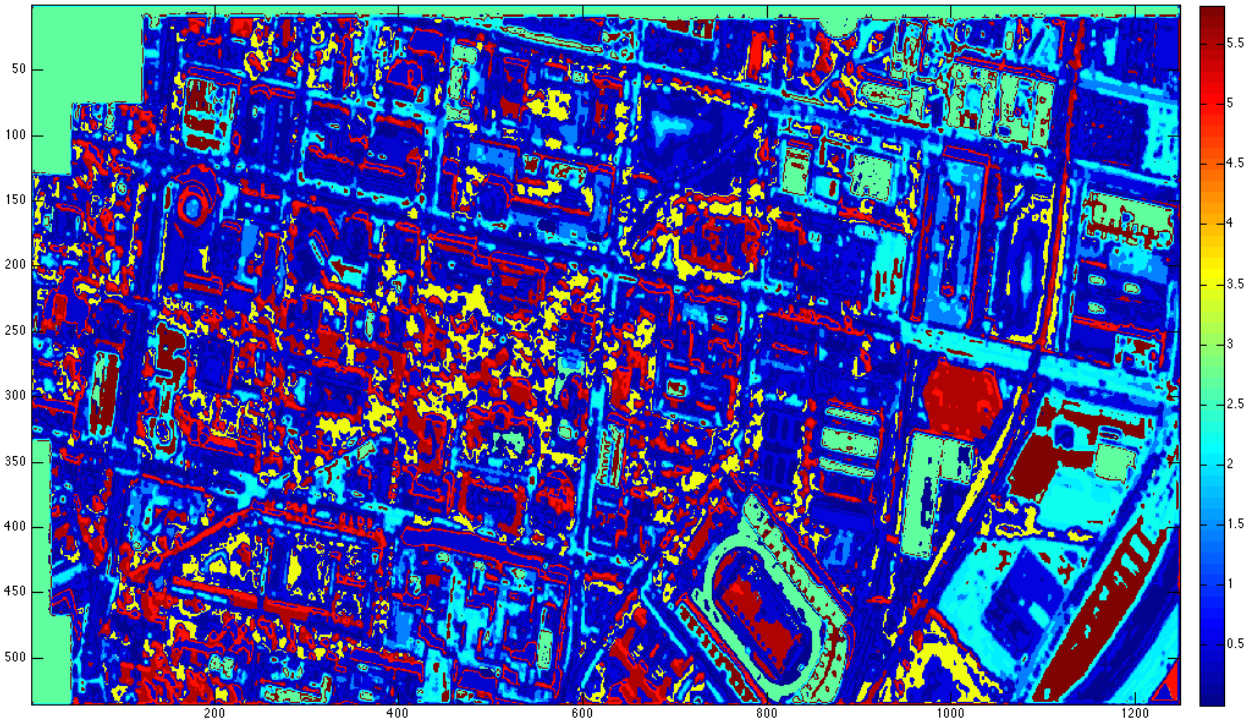

Figure 2: Driving costmap

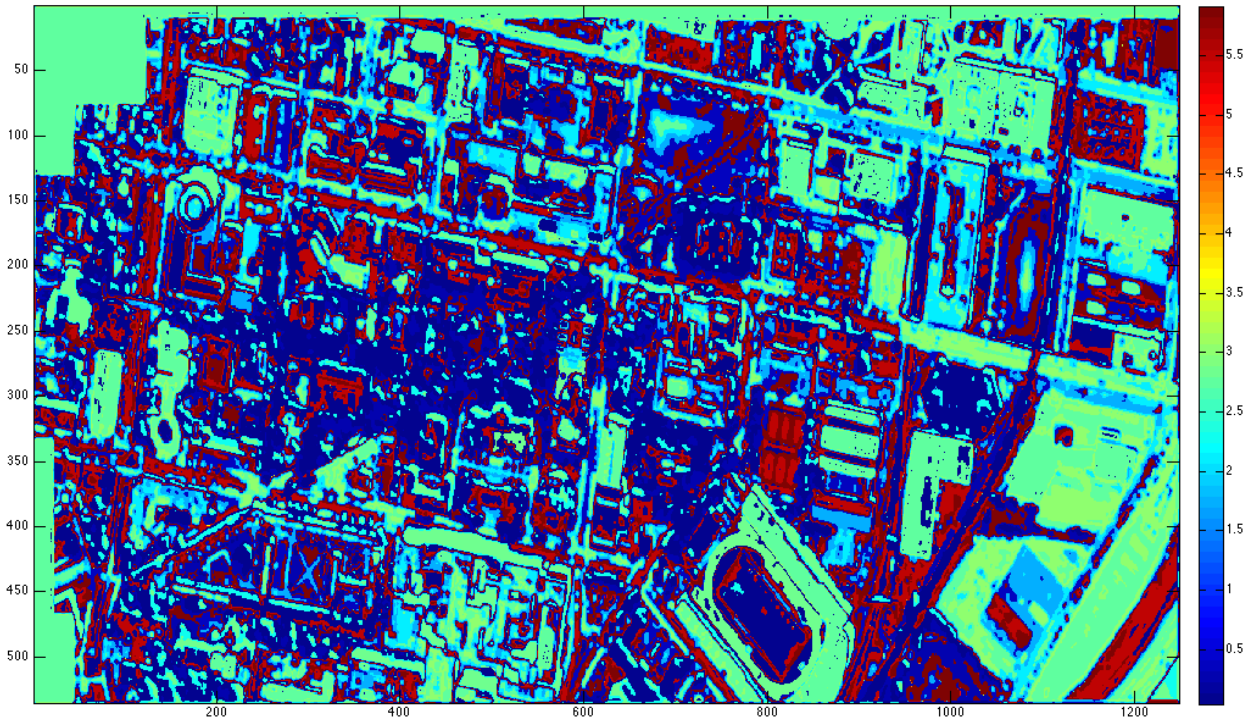

Figure 3: Walking costmap

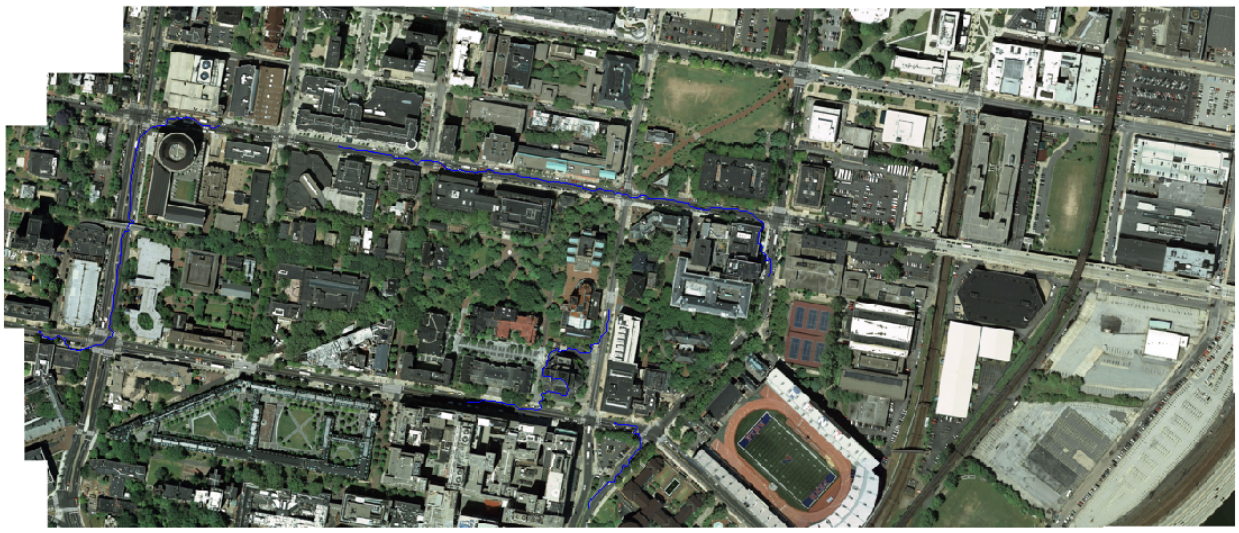

Figure 4: Planning examples for driving modality. Three examples are shown: A long path from west campus to the tennis courts, a medium-length path from the hospital to the Fischer library, and a short path around the triangular corner.

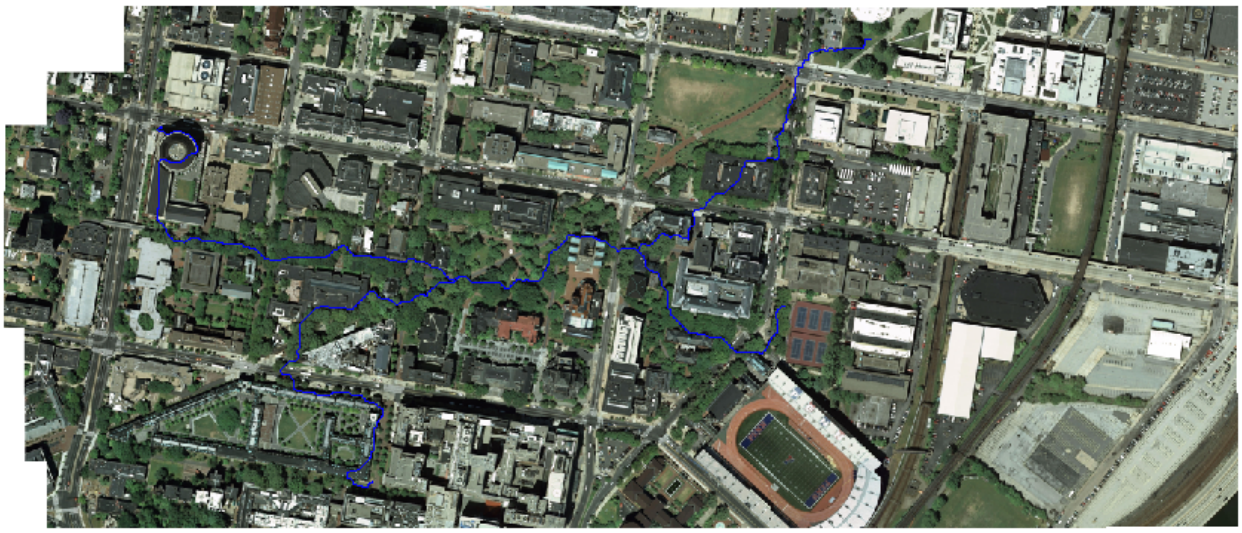

Figure 5: Planning examples for walking modality. Two examples are shown, wharton - to - drexel and hospital - to - tennis courts

Analysis and Conclusion

The algorithm performs fairly well for the driving modality and can plan long paths that stay on roads. However, because of the similar color of the roads and rooftops, the driving planner sometimes produces paths that go on the top of buildings (for example, the medium-length path in Figure 4.

The results in the walking modality tend to be less consistent than the driving modality. In my training set for walking, I allowed the pedestrian to walk anywhere that was not a building or a road. I think that because the training set was very diverse, the cost map was less discriminative than for driving, resulting in paths that primarily lay in the trees but sometimes crossed over roads and buildings as well.

Inspecting the cost maps (Figures 2 and 3), we see that the driving costmap is low cost (dark blue) on all the roads, but also dark blue on the tops of the dark buildings. Fortunately, the sidewalks show up as very high cost (red and yellow), and usually prevent the driving paths from straying off the roads. In the walking costmap, trees and grass have the lowest cost (dark blue), but many buildings and some stretches of road are fairly low-cost as well (light blue / green). As a result, the walking planner is less discriminative about crossing buildings and roads than we might like.

I think my results could have been made more consistent by expanding my feature set to include more sophisticated features. For example, edge or corner detection could have been used to classify buildings and roads more accurately, allowing my planner to avoid confusing buildings for roads. For the walking modality, I could have constructed my training set more carefully to include only areas where pedestrians can definitely walk, like sidewalks and footpaths. In the future, it would also be interested to see if cost augmentation would improve the quality of my results.

References

[1] Ratliff, N., Silver, D., & Bagnell, J. (2009). Learning to search: Functional gradient techniques for imitation learning. Autonomous Robots, 1–36. Retrieved from http://link.springer.com/article/10.1007/s10514-009-9121-3