Phone: (408) 840-1368

Scott Y. Shi

Email: scottyshi@gmail.com

http://scottyshi.github.io
https://github.com/scottyshi

Education

University of California – Los Angeles October 2014 – June 2018

- Pursuing B.S. in Computer Science -- Junior Year
- Cumulative GPA: **3.89**
- Completed Coursework: Operating Systems, Algorithms, Machine Learning (Coursera)
- In Progress Coursework: Artificial Intelligence, Database Systems, Convolutional Neural Networks for Computer Vision (Stanford CS 231n), Machine Learning (Statistics), Programming Languages, Distributed Machine Learning, Linear Models (Statistics)
- UPE Officer Board Member (Computer Science Honor Society)
- ACM Artificial Intelligence Committee Member

Skills

Programming languages: C++, C, Python, SQL, Matlab, R
Computer applications: Bash, Git

Experience

Software Engineering Intern: Bloomberg L.P. June 2016 – August 2016

- Configured relational database tables to collect necessary information and permissions to utilize.
- Integrated autocomplete on a big dataset into several Bloomberg terminal interfaces.
- Wrote offline tasks to maintain and regulate autocomplete as a real time service.

Undergraduate Research: Directed Research under Prof. Miodrag Potkojnak March 2016 – June 2016

- Developed R code to predict regions where active taxi trips would end up based on a big dataset.
- Utilized statistical methods to generate confidence intervals for the amount of expected customers and taxi drivers to pop up in any region for any specified time interval.
- Executed a bipartite matching of customer to taxi driver to minimize distance travelled in between rides.
- Minimized distance travelled on average by four times when comparing our theoretical results to the original results.

Undergraduate Research: UCLA Wireless Health Institute June 2015 - August 2015

- Developed Matlab code to utilize NSIM (open source Auditory Neural Network software that measures speech degradation through phone calls) to measure speech difference.
- Tested Validity of using NSIM as a means to measure speech difference by comparing its results against human perceived results

Projects

AI Tic Tac Toe: Self Study June 2016

- Utilized minimax algorithm to program an unbeatable AI for tic tac toe.
- Implemented alpha-beta pruning to speed up computation of game states.
- Created API to calculate AI moves in the backend while a human player plays in the front end.

FPGA Character Recognition: UCLA Logic Design Capstone Project May 2016

- Wrote and trained (with the MNIST dataset) a classifier to differentiate between four handwritten digits.
- Tuned the hypothesis function so it would be able to be calculated on an FPGA (no multiplication, no sigmoid function, restricted space).
- Implemented Verilog code to perform predictions on a variety of different inputs.