
Phone: (408) 840-1368

Scott Y. Shi
<http://scottyshi.github.io>
<https://github.com/scottyshi>

Email: scottyshi@gmail.com

Education

UCLA -- Junior in Computer Science

October 2014 – June 2018

- 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

- Wrote procedures in SQL for relational databases to collect entities and permissions information.
- Incorporated this big dataset into autocomplete to create a reusable front end search bar.
- Coded offline tasks to maintain and regulate the autocomplete search bar as a real time service.
- Worked with version control, code review, and code documentation on a regular basis.

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.
- Performed bipartite matching on customer to taxi driver based on closest distance.
- Reduced distance traveled 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 NSIM by comparing its results against human perceived results.

Projects

M-N-K Game AI: Self Study

June 2016

- Utilized minimax algorithm to program an unbeatable AI in Python for the M-N-K game.
- Allowed for a tunable difficulty of the AI on different dimensions of the board by controlling depth of computation of game states.
- Implemented alpha-beta pruning to speed up computation from several seconds to less than a second.

FPGA Character Recognition: UCLA Logic Design Capstone Project

May 2016

- Implemented logistic classifiers to differentiate between four handwritten digits using the NMIST dataset.
- 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.