Chih-Wei Chang

(213)-321-4914
cwchang@cs.cmu.edu
https://github.com/lazywei

Education

Carnegie Mellon University, School of Computer Science **Pittsburgh, PA**

M.S. Computational Data Science (MCDS) - Analytics Track (GPA: 3.78/4.0) Aug. 16 - Dec. 17

National Taiwan University **Taipei, Taiwan**

B.S. Mathematics (CS related GPA: 3.91/4.0, Last 60 GPA: 3.87/4.0) Sep. 11 - June 15

Skills

Language: Python, Ruby, Go, JavaScript, C/C++, CUDA, UNIX shell script, R, SQL

Tools: Linux, Git, Scikit-Learn, PyTorch, libsvm, AWS, Chef, Docker, MapReduce

Courses: Distributed Systems (ongoing), Parallel Computer Architecture and Programming, Machine Learning (PhD level), Machine Learning with Large Dataset, Search Engine, Intermediate Statistics, Advanced Multimedia Analysis, The Design and Analysis of Algorithms, Scientific Computing, High Dimensional Statistical Analysis and ML

Experience

Software Engineer Intern, Google, Mountain View, CA **May 2017 - Aug. 2017**

- Devised a new Z-score ensemble strategy to improve the quality of the conversions of Display Ads and increased the Conversion Per Dollar by 2% in the 1% traffic experiment.
- Designed a strategy to reconstruct the underlying scores distribution produced by different models based on the Maximum Likelihood Estimation (MLE) for Truncated Normal Distribution.
- Built a MapReduce fashion pipeline for logs extraction, data analysis, simulation, and MLE estimation.

Co-founder & CTO, Yoctol Info, Taipei, Taiwan **July 2014 - July 2015**

- Designed and built the entire data infrastructures, including servers, automated testing and deployment environment with CircleCI and Docker, RESTful APIs with Rails, and front-end web interfaces with ReactJS.
- Co-developed internal automated machine learning analytics tools based on SQL and relational database.

Full-stack Web Developer Intern, Codementor, Taipei, Taiwan **July 2014 - July 2015**

- Implemented APIs for both web and mobile; developed real-time online chatroom with Rails and ReactJS.
- Independently built a mobile app with React Native, including building back-end server and database support.

Back-end Web Developer Intern, Polydice Inc., Taipei, Taiwan **Feb. 2013 – July 2014**

- Revised internal auto deployment and scaling system with Chef and Amazon Web Services (AWS).
- Implemented service-oriented architecture and on-demand dynamic image resizing in Go.

Honors and Awards

The Alexa Prize, Amazon, Team CMU with selected sponsorship **Sep. 2016 - Sep. 2017**

- Built a non-task oriented conversational Artificial Intelligence on the Amazon Alexa.
- Implemented the neural dialogue generation module based on the seq-to-seq learning and topic detection.
- Employed character-level Recurrent Neural Network and made it context-aware through session management.

Predicting dropouts in MOOC (ACM KDD CUP 2015), Team NTU, 4th prize **Summer 2015**

- Contributed to feature generation and base models tuning in blending stage.
- Studied the AdaBoost for ranking (RankBoost) for Area Under Curve specific optimization.

Selected Projects and Open Source

ParGMRES: A Parallel Linear Solver **Parallel Programming Spring 17, Course Project**

- Implemented the parallel version General Minimal Residual (GMRES) algorithm using both OpenMP and CUDA.
- Analyzed the effects of using different matrix representation on performance under OpenMP parallelism; identified the bottleneck of parallelization in terms of an iterative algorithm.

Lucene-based Search Engine **Search Engine Fall 16, Course Project**

- Implemented a search engine in Jave from scratch; built essential query operators and retrieval methods, including Boolean match, Indri, and BM25, based on Lucene index.
- Developed several state-of-the-art extensions to the search engine, such as query expansion with pseudo relevance feedback, learning to rank via RankSVM, and re-ranking through diversification algorithms.

Mockingbird: Language classifier augmentation for Linguist **GitHub, Google Summer of Code 2015**

- Reduced memory usage and eliminated potential memory leak by rewriting the classifier from Ruby to Go.
- Implemented Go wrapper for LIBLINEAR C library to enable the use of different linear classifiers such as Logistic Regression and Linear SVMs; the wrapper source is released as “LIBLINEAR (Go wrapper)”.