JIANFENG CHEN

(919)-457-2034 ◇ https://jianfeng.us ◇ jchen37@ncsu.edu • ginfungchan@gmail.com

EDUCATION

PhD in Computer Science Aug 2014 - May 2019
North Carolina State University, GPA: 4.0/4.0
Dissertation: On the Value of Sampling and Pruning for Search-Based Software Engineering

Master in Computer Science *(En route)* Aug 2014 - Dec 2018
North Carolina State University, GPA: 4.0/4.0
Coursework: Data-to-Knowledge | DevOps | Advanced AI | Algorithm Analysis | Data Mining | Automated SE

BS in Computer Science Sep 2010 - May 2014
Shandong University, China, GPA: 89.93/100
Coursework: Data Structure | OS | Networking | Database System | Numerical Analysis | Image Processing

SKILLS AND INTERESTS

Language: Python, C++, Java, JavaScript, Matlab and SQL;
Data analysis tools: Scikit-learn, SciPy, Pandas, jMetal, Gephi;
DevOps tools: Jenkins, Ansible, Travis-CI, AWS Elasticsearch, S3, Docker, Redis.

INTERN AND RESEARCH EXPERIENCE

Spike: Predicting Breakdowns in Cloud Services Sep 2018- Apr 2019
Cooperation project with LexisNexis Legal & Professional

- Fetched cloud services monitoring data in microservices systems;
- Cleaned up the data and built machine learning models to predict the breakdowns 30 minutes ahead;
- Got a alarm system for service spikes with recalls and precision of 75% and above.

Cross-Stan: Embedding Bayesian Modeling in Python and C++ Programs May 2018- Aug 2018
Internship at Facebook (Machine learning experience group)

- Created the tool *CxStan* to embed Bayesian Modeling engine, Stan, into Python and C++ programs;
- Built a hierarchical framework to accelerate the Monte Carlo Sampling for the Stan modeling;
- Integrated the tool into the buck build tool, making the Stan engine as a service;
- Applied the *CxStan* to some Facebook services' traffic policing.

Automated Configurations for Cloud-based Workflows May 2017 - Aug 2017
North Carolina State University

- Presented a novel stochastic method for rapidly configuration cloud-based workflows;
- Automatically deployed the workflow with more than 500 sub-tasks to AWS platform. Save up to 30% economy cost within specific deadline requirement, compared to default greedy deployment policy in AWS.

LACE Data Privatization Tools and its Application Aug 2016 - Nov 2016
NSA funded project in RAISE Lab

- Distributed a data anonymization package in Python (see http://tiny.cc/pydp); tested package via Travis-CI;
- Applied my package to education and medical data sets. Evaluate data set utility through supervised learning.

Fast Principal-component-analysis (F-PCA) Method for Flight Status Log May 2016 - Aug 2016
Google Summer of Code program 2016

- Accepted by Google GSoC2016 program among 18,981 applicants **(accept rate: 6%)**;
- Hierarchical clustering a dataset(flight status log) with more than 20M entries top-down and bottom-up. Create a PCA-like dimension reduction algorithm and speed it up by spark. Compared my own algorithm with PCA.

Sampling vs. Searching in Search-based SE Dec 2014 - Aug 2017
North Carolina State University

- Created a sampling technique to solve the software product line problem. Deployed the algorithm into platform LSF by MPI; implemented a job schedule engine. Reduced the experiment time from 2 months CPU hrs into 11.5 hrs.
- Modeled the Linux Kernel modules in CNF sets. Created the decision tree surrogate model. By combining SAT solvers, found a way to configure large software systems 2000x faster. Published the results in TSE.

SELECTED COURSEWORK PROJECTS

Building Movie Recommendation System Aug 2015 - Dec 2015
"Netflix Prize" completion extension

- Built a movies recommendation system by training from 100 million Netflix ratings by Factorization Machine, SVM and ANN. Accelerated the learning process with HPC server.
- Crawled cast, critic reviews from rotten tomatoes and classified the movies basing on Jaro-Winkler Distance. Reduced the RMSE by up to 9% with the help of external information.

Continuous Integration/Delivery Pipeline Aug 2015 - Dec 2015
DevOps practice

- Basing on abstract syntax tree, created a regression test suite generator;
- Integrated Ansible scripts, Docker and Jenkins to build and deploy our "sunrise-calculator" app;
- Details at `http://tiny.cc/jc_devops`.

PUBLICATIONS

[1] Jianfeng Chen and Tim Menzies. "On the Benefits of Restrained Mutation: Faster Generation of Smaller Test Suites" Submitted to IEEE/ACM International Conference on Automated Software Engineering (ASE 2019).

[2] Jianfeng Chen, Joymallya Chakraborty, Philip Clark, Kevin Haverlock, Snehit Cherian and Tim Menzies. "Predicting Breakdowns in Cloud Services (with SPIKE)". Submitted to Symposium on the Foundations of Software Engineering (ESEC/FSE 2019 - Industry Paper Track)

[3] Jianfeng Chen, and Tim Menzies. "RIOT: A Stochastic-Based Method for Workflow Scheduling in the Cloud." 2018 IEEE 11th International Conference on Cloud Computing (CLOUD). IEEE, 2018. (Accept rate: 15%).

[4] Jianfeng Chen, Vivek Nair, Rahul Krishna, and Tim Menzies. "Sampling as a Baseline Optimizer for Search-based Software Engineering." IEEE Transactions on Software Engineering (2018).

[5] Jianfeng Chen, Vivek Nair, and Tim Menzies. "Beyond evolutionary algorithms for search-based software engineering." Information and Software Technology (2017).

[6] Junjie Wang, Song Wang, Jianfeng Chen, Tim Menzies, Qiang Cui, Miao Xie and Qing Wang. "Characterizing Crowds to Better Optimize Worker Recommendation in Crowdsourced Testing.". IEEE Transactions on Software Engineering(2019).

[7] Vivek Nair, Amrit Agrawal, Jianfeng Chen, Wei Fu, George Mathew, Tim Menzies, Leandro Minku, Markus Wagner, and Zhe Yu. "Data-Driven Search-based Software Engineering." The Mining Software Repositories (MSR) 2018.

[8] Tianpei Xia, Rahul Krishna, Jianfeng Chen, George Mathew, Xipeng Shen, and Tim Menzies. "Hyperparameter optimization for effort estimation." Submitted Empirical Software Engineering (EMSE) 2018

[9] Vivek Nair, Tim Menzies, and Jianfeng Chen. "An (accidental) exploration of alternatives to evolutionary algorithms for SBSE." In International Symposium on Search Based Software Engineering, pp. 96-111. Springer, Cham, 2016.