Fei Liang

6158 Waterman Blvd, St. Louis MO, 63112
Tel: (314) 435-5113; Email: liangfei@wustl.edu

EDUCATION

Master of Computer Science

Washington university in St. Louis, St. Louis MO, USA September 2015 – May 2017

- Concentration: Software Development, Data Science
- Relevant Coursework: Advanced Algorithms, Machine Learning, Data Mining, Bayesian Methods in Machine Learning, Multi-Agent Systems, Cloud Computing with Big Data Applications, Database Management Systems, Video Game Programming, Mobile Application Development, Introduction to Artificial Intelligence
- Cumulative GPA: 3.40/4
- Certification: Certificate in Data Mining and Machine Learning

PROJECT EXPERIENCE

Image Classification of Dogs and Cats using Gaussian Process

Washington University in St. Louis April, 2017 – May, 2017

Performed classification on labeled images containing dogs or cats with Gaussian Process

- Wrote scripts for the team to formalize the data from multiple sources into different formats.
- Implemented principal component analysis and singular value decomposition to extract features.
- Enhanced classification's accuracy with Bayesian linear regression from 0.541 to 0.869 compared with linear regression

Environment: Python, Matlab

iOS Application Development

Washington University in St. Louis October, 2016 – December, 2016

An iOS and Firebase based application that allows users to give and receive ratings from others.

- Designed the framework and contributed more than 95% of the codes for the application.
- Accomplished syncing and storing data between clients and Realtime Database as well as Firebase Storage.
- Developed GPS service based on Mapkit and CoreLocation that let users interact with others.
- Implemented custom View Controllers like Table View, and developed simulated local notification.

Environment: Xcode, Firebase Service, Swift, Git

Clustering and Classification for Gene Functions

Washington University in St. Louis September, 2016 – December, 2016

Applied methods from machine learning and data mining to cluster genes with same function and make predictions.

- Used 7-NN to impute missing data; applied PCA, SVM and information gain to select features.
- Built models with random forest, neutral network, K-means, hierarchical clustering and some other methods.
- Achieved result: 86% accuracy for classification, 62% accuracy for clustering.

Environment: Weka, Matlab, Python, Scikit

Recommendation System Based on Netflix Rating Dataset

Washington University in St. Louis April, 2016 – May, 2016

Developed a system to recommend movies to users based on data of user-movie ratings and collaborative filtering.

- Implemented MapReduce process via python and Hadoop streaming to generate normalized movie-movie model and user-user model from Netflix Rating Dataset
- Participated in implement Pearson Correlation as similarity measure using another two MapReduce jobs.
- Applied Top-K and weighted average on similarities to make predictions.

Environment: MapReduce, Hadoop, Pig, Matlab, Python

3D Video Game Development

Washington University in St. Louis February, 2016 – April, 2016

A video game that players can control the spaceship to dodge intense bullets and attack enemies.

- Made models for the spaceship and part of game items in Blender; created scenes for three levels in Unity3D.
- Wrote scripts to randomly generate trees in the background; implemented codes for physics effects.
- Contributed creative ideas and accomplished extra game elements and mechanisms like storylines.
- Involved in the playtest in St. Louis Science Center and collected feedback from people of different ages and jobs.

Environment: Unity3D, Blender, C#, Git

Design of Image Segmentation Algorithm based on CUDA

Central South University January, 2014 – June, 2016

Deployed CUDA on GPU to speed up massive images' segmentation using Otsu's method

- Developed parallel image processing algorithms based on classic Otsu's method
- Implemented algorithms for classic and parallel Otsu' methods via C++ and CUDA C.
- Tested the image segmentation program on GPU; verified the efficiency comparing with classic Otsu's method

Environment: Visual Studio, C++, CUDA C**,** OpenCV

SKILLS

- **Language:** Python(proficient), Swift, SQL, Java, C++, C#
- **Software:** Matlab, Xcode, Visual Studio, Jupyter Notebook, Sublime, Postgres, Pig, Weka, Git, MapReduce, Hadoop