Tejas Lodaya

tlodaya@purdue.edu
https://tejaslodaya.com
+1-765-409-0686

Graduate Student with interests in systems, storage and ML. Looking for **internships** starting **May 2022**

Education

- **Purdue University** — West Lafayette, Indiana
  M.S in Computer Science (advised by Dr. Jianguo Wang) — *2023*
  Ongoing work: Improving index construction in vector databases for billion-scale similarity search
- **PES Institute of Technology** — Bangalore, India
  B.E in Computer Science; GPA: 9.00/10.0 (Class Rank: 9) — *2017*

Professional Experience

- **Swiggy** — Bangalore, India
  Senior Software Engineer — *May 2019 - Aug 2021*
  - Designed end-to-end platform to serve deep learning models at scale using tensorflow serving with 50+ models in production. Attained p99 latencies of 15ms at peak 10k RPS with sidecar containers and gRPC network calls
  - Maintained and operated ML platform with over 5k feature jobs and ~150 scala models in production
  - Streamlined visibility and monitoring of ML models by feature pipeline quality checks, alerts on data drifts, erroneous models, and infrastructure failures. Reduced TAT for model failures from 1 day to 2 hours
  - Migrated 900+ production jobs to databricks as a result of complete spark infrastructure outage (S1 severity) in 7 hours. No degradation in business metrics or model prediction accuracy to any consumers of runtime
  - Developed map-matching algorithm to snap driver GPS pings to underlying road network. Powering accurate driver payouts saving 12p/order, missing road detection contributing 1300 roads back to public OSM and ETA predictions
  - Architected geospatial querying engine for OLAP-style data analytics, with average latencies of 3 mins for point-in-polygon queries and ingestion rate of 10k/sec with parallel indexing, date partitioning and compression
  - Built a million-scale approximate nearest neighbour search using vector similarity to discourage fraudsters from reusing images to claim refund. Obtained latencies of 250ms at peak 3k RPS using ES, social graph and caching
  - Engineered a low-throughput, high-latency prototyping platform to serve python-based models at scale. Unlocked data scientists to conduct experiments and validate hypothesis without rewrite in high-performant scala/tensorflow
  - Deployed smart payments model into checkout springboot service with a peak throughput of 2mn requests per day. Generated real-time features from order flow and coded up JUnit cases for compile-time testing
  - Reduced AWS cost by 25% as part of operational excellence by optimizing spark jobs and fine-tuning clusters
  - Devised a performance testing framework to measure how fast ML models can fetch features and produce results in critical software systems. Reduced number of PDs by 70% and gauged model readiness and latency in production
  - Oversaw design and solutioning of platform's retraining capability, orchestration framework and feature store
  - Established a team of 4 engineers (2 SDE2, 2 SDE1) and supported operational tasks of 40 data scientists by setting up an on-call process. Conducted 100+ on-site design interviews for SDEs and engineering rounds for DS
- **Noodle.ai** — Bangalore, India
  Software Engineer — *Jul 2017 - May 2019*
  - Migrated legacy codebase from in-memory redis cluster to cassandra, reducing costs by $250k per year. Decoupled code from data-model, implemented connection pooling and integrated memcached to reduce latency by 30%
  - Conceptualized and orchestrated demand forecasting ecosystem with MySQL and airflow to run batch jobs in R, bringing down manual effort by 10 hours. Scaled compute using SPMD on N cores to reduce latency by 76%
  - Formulated an incremental learning framework by an ensemble of global long-term model and local short-term model. Reduced retraining frequency from daily to weekly at marginal loss of accuracy

Open Source Contributions

- **Variational Recurrent Autoencoder**: Unsupervised, feature-based time-series clustering algorithm in pytorch
- **Troop**: Perform chunkwise data.frame processing across multiple cores on SNOW clusters with low memory footprint

Skills

- **Languages**: Python, Java, Golang, C++, R
- **Datastores**: Cassandra, Postgres, DynamoDB, Redis, ElasticSearch, Kafka (as a data-store)
- **Frameworks**: Spark, Tensorflow, Airflow, Flink, Hive, Presto, Snowflake
- **Misc.**: GeoMesa, AWS, GCP