Regressing Towards Simpler Prediction Systems

Sibyl project, Google Inc
(Speaker: Tushar Chandra)
Example: YouTube recommendations

Recommended Videos
Prediction system architecture

Analysis tools

Server

User

Machine learned model

"Joined" logs

Machine learning system

Impression log

Interaction log

... etc ...

Databases
What is Sibyl?

- Large-scale machine learning system
  - Classification, regression, factorized models
- Designed for the largest Google data sets
  - Process 100B+ training examples
  - Explore 100B+ unique features
  - Serve models with 1B+ features
- Uses principled, state-of-the-art algorithms
What is Sibyl?

- Prediction platform
- Widely used across Google
Why Do Teams Collaborate with Sibyl

● Because we help teams build simple, maintainable, understandable prediction systems that significantly improve their product

● Not because of our machine learning algorithms
Wins from ML

Manual System
  ↓ +50%
Add Machine Learning
  ↓ +10%
Good Machine Learning
  ↓ +2%
Exceptional Machine Learning

Wins from Features

Document Features +5%
Query Features +5%
UI Features +5%
\(O(10)\) such +5% each
How Sibyl Started

Good Machine Learning

+2% (business metrics)

Exceptional Machine Learning

- Competed with existing system
  - +1.5%, cost of switching was too high
Case Study: Small YouTube property

- Only two feature types (Country, Video ID)
- 1 month to implement linear regression
- 9 months to launch
- 20% improvement

Machine Learning

+10% (business metrics)

Good Machine Learning
Config

How users use the system

Started out as an after-thought

Now
- On version 4.5
- >40K lines of code in the core
- ~100K lines of user code
- 2 person dev team and growing

Several other machine learning platform teams have had similar experiences
Config challenges

Orchestrate all components in the system, e.g.,
```
c.Monitoring.SetAlertRecipients("tusharchandra@google.com")
```

Setup models for training
- Transform data (e.g., discretize and train on bucket-id)
- Specify features, loss function, hyperparameters, etc.

Used Go as our config language
- Better type system than Python
Research instinct: add capability

Business as usual
(Add capability)
Actively manage complexity

- Simplify
- Thoughtfully add capability
- Business as usual (Add capability)
Thoughtfully Adding Factorization

1. Factorization → Clustering → Regression
2. Factorization + Clustering → Regression
3. Factorization + Regression

Prototype → Integration → Final Product
Future directions

New deployments

- Make it much easier
  - System refinement
  - Model understanding
  - Education
  - 10x scale

Mature deployments

- Feature engineering
  
Manual feature engineering has run its course
Need new algorithms, e.g.,
  - Neural nets
  - Ranking
  - Embeddings
  - Hidden state
A note on Sibyl’s performance

- Based on MapReduce and Google’s cluster file system
- Iterative batch learner
- Spends most of its time on ML computation
  - Less than 50% of its time on MR+GFS+I/O

Conclusion: MR + cluster file system is a reasonable abstraction (from a performance perspective)
- See invited talk at DSN 2014 (on YouTube)