
LinkedIn

GLMix: Generalized Linear Mixed Models For Large-Scale Response Prediction

XianXing Zhang

Yitong Zhou

Yiming Ma

Bee-Chung Chen

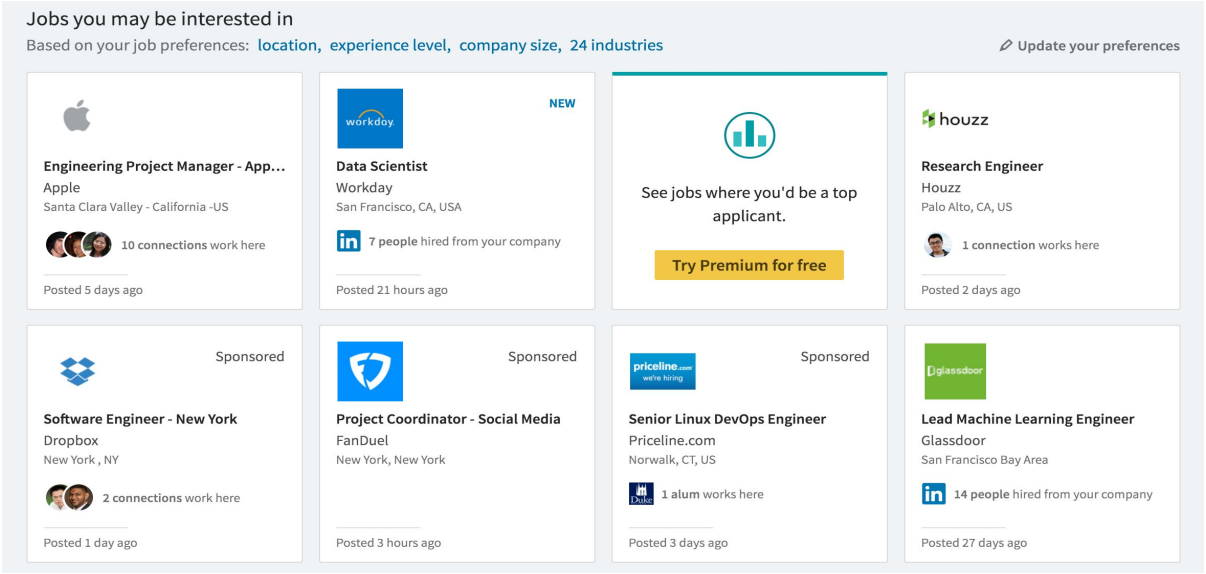
Liang Zhang

Deepak Agarwal

{xazhang, yizhou, yma, bchen, lizhang, dagarwal}@linkedin.com

GLMix @

- Jobs homepage
  - **+20-30%** lift in job application rate
  - Ramped to serve **100%** traffic (400 million members)



GLMix @ 

- Jobs homepage
  - **+20-30%** lift in job application rate
  - Ramped to serve **100%** traffic
- Article recommendation
  - **+10-20%** lift in CTR
- Sponsored update (Ads)
  - Reduces model training time by **10x**
- People viewed this also viewed
  - **+10-20%** lift in offline experiments

GLMix: a **production-ready** model that **works**

Why GLMix

Desired model properties

- **Interpretable** with features

Desired model properties

- **Interpretable** with features
- **Scalable** with the economic graph

Desired model properties

- **Interpretable** with features
- **Scalable** with the economic graph
- **Deployable** with existing (e.g., open-sourced) infras

Desired model properties

- **Interpretable** with features
- **Scalable** with the economic graph
- **Deployable** with existing (e.g., open-sourced) infras
- **Fine-grained** with users or items with more data

Generalized Linear Model (GLM)

- Predicting the response of ***user i*** on ***item j***:

$$g(\mathbb{E}[y_{ij}]) = \mathbf{x}'_{ij}\mathbf{w}$$

- $\mathbf{x}_{ij}$: Feature vector
- $\mathbf{w}$: Coefficient vector
- $\mathbb{E}[y_{ij}]$: Expectation of response
- $g(\cdot)$: Link function

Generalized Linear Model (GLM)

- Predicting the response of ***user i*** on ***item j***:

$$g(\mathbb{E}[y_{ij}]) = \mathbf{x}'_{ij}\mathbf{w}$$

- Interpretable ✓
- Scalable ✓
- Deployable ✓

Generalized Linear Model (GLM)

- Predicting the response of ***user i*** on ***item j***:

$$g(\mathbb{E}[y_{ij}]) = \mathbf{x}'_{ij}\mathbf{w}$$

- Interpretable ✓
- Scalable ✓
- Deployable ✓
- Fine-grained ?

An example in job recommendation

$$g(\mathbb{E}[y_{ij}]) = \mathbf{x}'_{ij}\mathbf{w}$$

User	Feature	Job	Company	Response
Alice	$\mathbf{x}_{\text{alice}}$	Data Scientist	Microsoft	✓
Alice	$\mathbf{x}_{\text{alice}}$	Data Scientist	airbnb	✗
Annie	$\mathbf{x}_{\text{annie}}$	Data Scientist	Microsoft	✗
Annie	$\mathbf{x}_{\text{annie}}$	Data Scientist	airbnb	✓

An example in job recommendation

$$\mathbf{x}_{\text{alice}} \approx \mathbf{x}_{\text{annie}} \Rightarrow \mathbf{w} \approx \mathbf{0}$$

User	Feature	Job	Company	Response
Alice	$\mathbf{x}_{\text{alice}}$	Data Scientist	Microsoft	✓
Alice	$\mathbf{x}_{\text{alice}}$	Data Scientist	airbnb	✗
Annie	$\mathbf{x}_{\text{annie}}$	Data Scientist	Microsoft	✗
Annie	$\mathbf{x}_{\text{annie}}$	Data Scientist	airbnb	✓

$$g(\mathbb{E}[y_{ij}]) = \mathbf{x}'_{ij}\mathbf{w}$$

Matrix Factorization (MF)

- Predicting the response of ***user i*** on ***item j***:

$$g(\mathbb{E}[y_{ij}]) = \mathbf{u}_i' \mathbf{v}_j$$

- $\mathbf{u}_i$: User latent factor
- $\mathbf{v}_j$: Item latent factor
- $\mathbb{E}[y_{ij}]$: Expectation of response
- $g(\cdot)$: Link function

Matrix Factorization (MF)

- Predicting the response of ***user i*** on ***item j***:

$$g(\mathbb{E}[y_{ij}]) = \mathbf{u}_i' \mathbf{v}_j$$

- Interpretable ? (Does not use features)
- Scalable ✓
- Deployable ✓
- Fine-grained ✓

GLMix: Generalized Linear Mixed Model

- Predicting the response of ***user i*** on ***item j***

$$g(\mathbb{E}[y_{ij}]) = \mathbf{x}'_{ij}\mathbf{w} + \mathbf{x}'_j\boldsymbol{\alpha}_i + \mathbf{x}'_i\boldsymbol{\beta}_j$$

- Model coefficients with different **granularities** / **dimensions**
  - **Per-user** coefficients $\boldsymbol{\alpha}_i$
  - **Per-item** coefficients $\boldsymbol{\beta}_j$

GLMix: Generalized Linear Mixed Model

- Predicting the response of ***user i*** on ***item j***

$$g(\mathbb{E}[y_{ij}]) = \mathbf{x}'_{ij}\mathbf{w} + \mathbf{x}'_j\boldsymbol{\alpha}_i + \mathbf{x}'_i\boldsymbol{\beta}_j$$

- Model coefficients with different **granularities** / **dimensions**
  - **Per-user** coefficients $\boldsymbol{\alpha}_i$
  - **Per-item** coefficients $\boldsymbol{\beta}_j$

- GLMix = GLM + per-user model + per-item model

GLMix for Job Recommendation

$$g(\mathbb{E}[y_{ij}]) = \mathbf{x}'_{ij}\mathbf{w} + \mathbf{x}'_{j}\boldsymbol{\alpha}_i + \mathbf{x}'_{i}\boldsymbol{\beta}_j$$

- Global model
  - Similarity between member profile and jobs profile, e.g. do the member skills and job skills look similar?

GLMix for Job Recommendation

$$g(\mathbb{E}[y_{ij}]) = \mathbf{x}'_{ij}\mathbf{w} + \mathbf{x}'_{j}\boldsymbol{\alpha}_i + \mathbf{x}'_{i}\boldsymbol{\beta}_j$$

- Global model
  - Similarity between member profile and jobs profile, e.g. do the member skills and job skills look similar?
- Per-member model
  - E.g. If a member has applied to a job with title = “software engineer”, we will boost “software engineer” jobs more in her results.

GLMix for Job Recommendation

$$g(\mathbb{E}[y_{ij}]) = \mathbf{x}'_{ij}\mathbf{w} + \mathbf{x}'_{j}\boldsymbol{\alpha}_i + \mathbf{x}'_{i}\boldsymbol{\beta}_j$$

- Global model
  - Similarity between member profile and jobs profile, e.g. do the member skills and job skills look similar?
- Per-member model
  - E.g. If a member has applied to a job with title = “software engineer”, we will boost “software engineer” jobs more in her results.
- Per-job model
  - E.g. If a job gets an apply with a member titled “software engineer”, we will boost this job more for members with this title.

Alice and Annie's problem revisited

$$g(\mathbb{E}[y_{ij}]) = \mathbf{x}'_{ij}\mathbf{w} + \mathbf{x}'_j\boldsymbol{\alpha}_i + \mathbf{x}'_i\boldsymbol{\beta}_j$$

User	Feature	Job	Company	Response
Alice	$\mathbf{x}_{\text{alice}}$	Data Scientist	Microsoft	✓
Alice	$\mathbf{x}_{\text{alice}}$	Data Scientist	airbnb	✗
Annie	$\mathbf{x}_{\text{annie}}$	Data Scientist	Microsoft	✗
Annie	$\mathbf{x}_{\text{annie}}$	Data Scientist	airbnb	✓

Alice and Annie's problem revisited

$$g(\mathbb{E}[y_{ij}]) = \mathbf{x}'_{ij}\mathbf{w} + \mathbf{x}'_j\boldsymbol{\alpha}_i + \mathbf{x}'_i\boldsymbol{\beta}_j$$

User	Feature	Job	Company	Response
Alice	$\mathbf{x}_{\text{alice}}$	Data Scientist	Microsoft	✓
Alice	$\mathbf{x}_{\text{alice}}$	Data Scientist	airbnb	✗
Annie	$\mathbf{x}_{\text{annie}}$	Data Scientist	Microsoft	✗
Annie	$\mathbf{x}_{\text{annie}}$	Data Scientist	airbnb	✓

$\boldsymbol{\alpha}_{\text{Alice}}$

$\boldsymbol{\alpha}_{\text{Annie}}$

GLMix model properties

- Interpretable ✓ (GLMix = GLM + per-user + per-item + ...)
- Scalable
- Deployable
- Fine-grained ✓

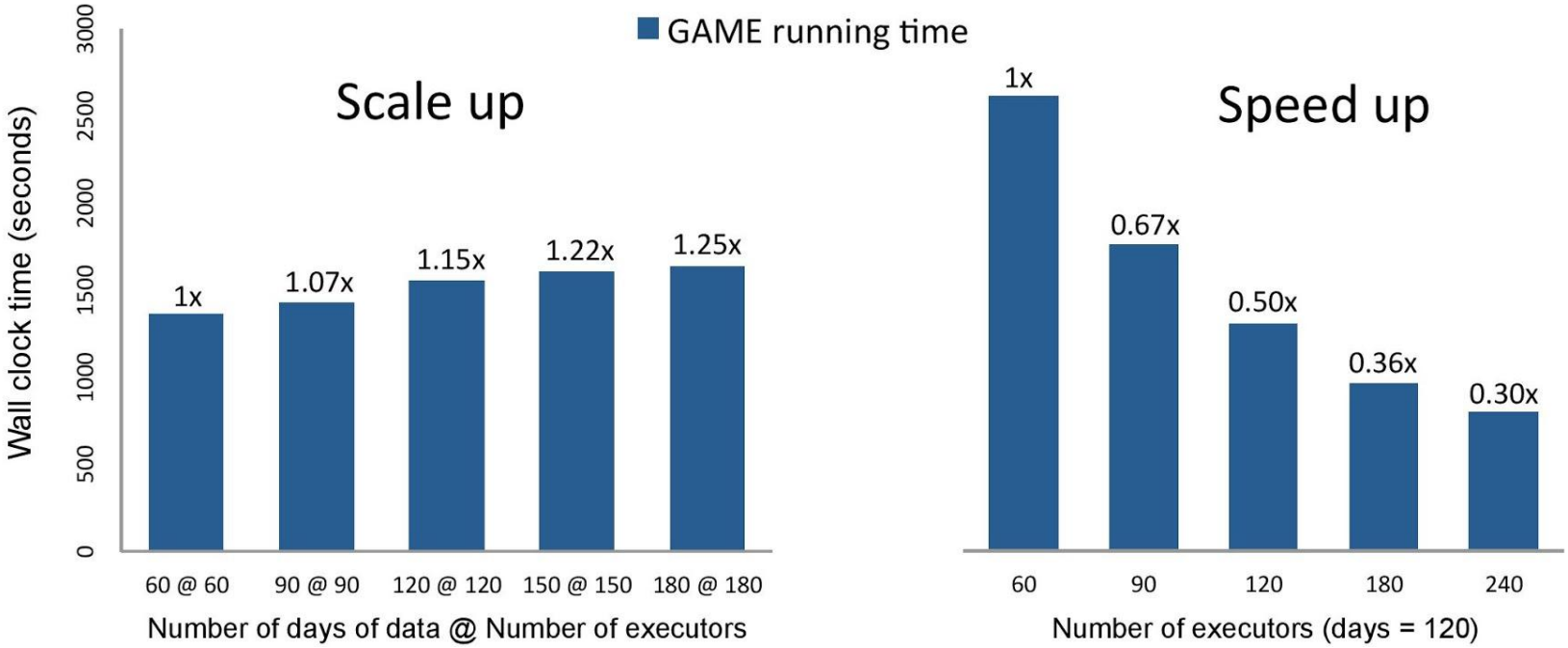
- Scalability challenges
  - 10 million users, 1000 features -> 10B coefficients
  - More users, more features?

- Scalability challenges
  - 10 million users, 1000 features -> 10B coefficients
  - More users, more features?

- Solutions (sketch)
  - Locality aware data pre-partitioning
    - Sharding data by both **rows** (data samples) and **columns** (features)

- Scalability challenges
  - 10 million users, 1000 features -> 10B coefficients
  - More users, more features?

- Solutions (sketch)
  - Locality aware data pre-partitioning
    - Sharding data by both **rows** (data samples) and **columns** (features)
  - Minimize communication cost
    - Coordinate descent, only residuals are shuffled


GAME running time
Scale up
Wall clock time (seconds)
0
500
1000
1500
2000
2500
3000
1x
1.07x
1.15x
1.22x
1.25x
60 @ 60
90 @ 90
120 @ 120
150 @ 150
180 @ 180
Number of days of data @ Number of executors
Speed up
1x
0.67x
0.50x
0.36x
0.30x
60
90
120
180
240
Number of executors (days = 120)

GLMix model properties

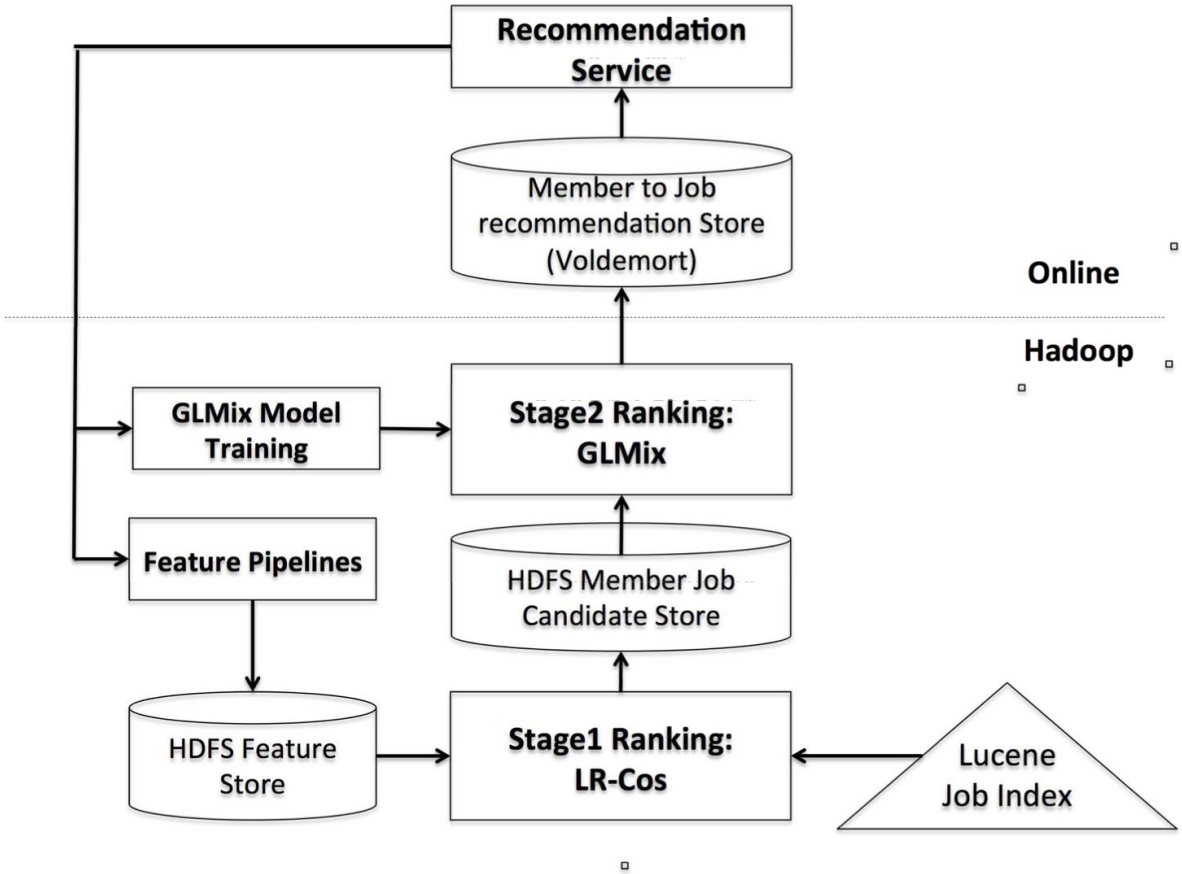
- Interpretable ✓
- Scalable ✓
- Deployable
- Fine-grained ✓

How to **deploy** GLMix with existing (open-sourced) infra?

- Offline scoring
  - Prepare and score the user-item candidate set offline
  - Push scores to online K-V store
  - Key: ***throughput***

- Online scoring
  - Store model coefficients online
  - Generate and score item candidate set for users in real time
  - Key: ***Latency***

Offline Scoring



GLMix @ 

- Jobs homepage
  - Ramped to serve 100% traffic
    - With offline scoring, GLMix is ramped to **8%** traffic
  - +20-30% lift in job application rate

Online Scoring for GLMix

Checkout our hands-on tutorial

“Building End-to-End Recommender System with Photon-ML”

- Keyword for Google search: “***Photon-ML***”
- https://github.com/linkedin/photon-ml/wiki/Photon-ML-Tutorial

GLMix @ 

- Jobs homepage
  - Ramped to serve 100% traffic
    - With offline scoring, GLMix is ramped to **8%** traffic
    - Ramped to serve **100%** traffic **with online scoring**
  - +20-30% lift in job application rate

GLMix model properties

- Interpretable ✓
- Scalable ✓
- Deployable ✓
- Fine-grained ✓

Takeaways

- **GLMix** is …
  - Interpretable
    - GLM + per-dimension models
  - Scalable
    - Sharded by samples and features + Coordinate descent
  - Deployable
    - With existing (open source) infra
  - Fine-grained
    - Per-dimension models

Takeaways

- **GLMix** is…
  - Interpretable
  - Scalable
  - Deployable
  - Fine-grained
  - **An open-source library**
    - Search for Photon-ML
    - https://github.com/linkedin/photon-ml
    - Implemented in the GAME module