ML at Twitter: A Deep Dive into Twitter’s Timeline

Cibele Montez Halasz, Twitter Cortex
Satanjeev “Bano” Banerjee, Twitter Timelines Quality

O’Reilly AI Conference
1. ML at Twitter
2. Timelines Product
3. Modeling and Serving Pipeline
4. Model
5. Timelines on Tensorflow
6. Lessons Learned and Conclusions
ML at Twitter
Personalized Search
Recommendations

Who to follow

- SirWarwick
  @SirWarwickLoL
- Yiyin
  @Yiyinslstr_
- Volodymyr Zhabiuk
  @vzhabiuk

Show more
Health

Counterchekist @counterche... · 4h

This Tweet is not available because it includes potentially sensitive content.

Replying to @counterchekist

Tweet your reply
Marvel Studios
@MarvelStudios
“You know your teams, you know your missions.”

Marvel Studios’ #AvengersEndgame is in theaters April 26. Get tickets now: Fandango.com/AvengersEndgame
Timelines Product
Twitter “Timeline”

Tweets shown to users when they log in and ask for Tweets in the Home section of the app

Similar to Facebook News Feed, Instagram Home, etc.
‘Reverse Chron’ Timeline

1 minute old

5 minutes old

9 minutes old

13 minutes old

29 minutes old
Problems of Reverse Chron

- The most recent Tweet may not be the most interesting Tweet
- The most interesting Tweets for you may be
  - From hours ago
  - From people who tweet infrequently
  - From people you don’t even follow
- Some Tweets may be bad: e.g. abusive, or near-duplicate content
Goal of Timelines Product

Show each user the Tweets they are most likely to be interested in
User interests —> User engagement

Engagement == Likes / replies / retweets / ...

Timelines Product’s Goal:

Show users Tweets that they are more likely to engage with
Relevance Ranked Timeline

- 29 minutes old
- 5 minutes old
- 13 minutes old
- 1 minutes old
- 9 minutes old
ML problem: Given a Tweet and a user, predict the user’s probability of engaging with that Tweet
Modeling and Serving Pipeline
Modeling and Serving Pipeline

CLIENT

request Tweets

TWEET SERVICE
Modeling and Serving Pipeline

CLIENT

request Tweets

TWEET SERVICE

fetch Tweets request

CANDIDATE TWEETS

request Tweets
Modeling and Serving Pipeline

CANDIDATE TWEETS

fetch Tweets request

Tweets

request Tweets

TWEET SERVICE

CLIENT
Modeling and Serving Pipeline

CANDIDATE TWEETS

fetch Tweets request

Tweets

TWEET SERVICE

score Tweets request

PREDICTION SERVICE

CLIENT

request Tweets
Modeling and Serving Pipeline

CANDIDATE TWEETS

fetch Tweets request

TWEET SERVICE

Tweets

request Tweets

score Tweets request

Tweets with scores

CLIENT

PREDICTION SERVICE

TWEETS

request

23
Modeling and Serving Pipeline

CANDIDATE TWEETS

fetch Tweets request

request Tweets

sort, threshold, return Tweets to user

TWEET SERVICE

Tweets

score Tweets request

Tweets with scores

PREDICTION SERVICE

TWEET SERVICE

client

request Tweets

Tweets

fetch Tweets request

client
Modeling and Serving Pipeline

CLIENT

fetch Tweets request

CANDIDATE TWEETS

fetch Tweets request

TWEET SERVICE

request Tweets

sort, threshold, return Tweets to user

score Tweets request

Tweets with scores

PREDICTION SERVICE

save as future training examples

TRAINING DATA

Tweets with scores

TRAINING DATA

Tweets

PREDICTION SERVICE

TWEET SERVICE

CANDIDATE TWEETS
Modeling and Serving Pipeline

CANDIDATE TWEETS

fetch Tweets request

TWEET SERVICE

Tweets

sort, threshold, return

Tweets to user

request Tweets

PREDICTION SERVICE

score Tweets request

Tweets with scores

save as future training examples

TRAINING DATA
Modeling and Serving Pipeline

CANDIDATE TWEETS

fetch Tweets request

TWEET SERVICE

request Tweets

Tweets

sort, threshold, return

Tweets to user

save as future training examples

PREDICTION SERVICE

score Tweets request

Tweets with scores

TRAINING DATA
Modeling and Serving Pipeline

CANDIDATE TWEETS

fetch Tweets request

Tweets

tweet service

request Tweets

sort, threshold, return

Tweets to user

PREDICTION SERVICE

score Tweets request

Tweets with scores

save as future training examples

TRAINING DATA

save engagements as labels
Modeling and Serving Pipeline

**CANDIDATE TWEETS**
- fetch Tweets request
- Tweets

**TWEET SERVICE**
- request Tweets
- sort, threshold, return Tweets to user
- score Tweets request
- Tweets with scores

**PREDICTION SERVICE**
- retrain engagement models

**TRAINING DATA**
- save as future training examples
- save engagements as labels

**CLIENT**
- request Tweets
- Tweets
- save as future training examples
- save engagements as labels
Total Daily Monetizable Daily Active Users

Source: Twitter 2018 Q4 Earnings Report
Modeling Engagements
Modeling Engagements
Modeling Engagements
Modeling Engagements
Modeling Engagements
Modeling Engagements
Modeling Engagements
Combining Separate Engagement Scores per Tweet

Currently we combine the models with learned weights:

\[
\text{score} = x \times P(\text{like}) + y \times P(\text{retweet}) + z \times P(\text{reply}) + \ldots
\]

Additional score adjustments:
- Photo penalty
- Author diversity penalty
- Producer-side value boost

Lots of experiments to tune these parameters.
Predictive Features for Timelines Ranking

➔ Tweet features

# 🔢 ❤️ 🗣

➔ User’s network features

➔ User historical features

➔ Request features

➔ Text features
Model
The Model

Timelines Model

1. Percentile Discretizer
2. Sparse Layer
3. Dense MLP
4. Isotonic Calibration
Timelines Model: Percentile Discretization
Timelines Model

1. Percentile Discretizer
2. Sparse Layer
3. Dense MLP
4. Isotonic Calibration
The Model

Timelines Model: Sparse Linear Layer

\[ N_j = F(\sum W_{i,j} \cdot \text{norm}(V_i) + B_j) \]

![Diagram of Timelines Model: Sparse Linear Layer](image)
Timelines Model

1. Percentile Discretizer
2. Sparse Layer
3. Dense MLP
4. Isotonic Calibration
Timelines Model

User Features Split Group

User Embedding

User-Author Split Group

U-A Embedding

Tweet Features Split Group

Tweet Embedding

tf.concat

Dense Neural Network

Objective
Timelines Model

1. Percentile Discretizer
2. Sparse Layer
3. Dense MLP
4. Probability Calibration
Timelines Model: Skewed Dataset

- Very few positive examples of likes, replies, retweets vs negative examples (Tweets that were shown to the user, but the user did not like, reply, retweet, etc).

- Solution:
  - Heavily downsample negative examples
  - Train without example weights
  - Then apply probability calibration in the end
Sparse Linear Layer: Probability Calibration

Goal: Bring predicted CTR back in line with data CTR
Sparse Linear Layer: Probability Calibration

MLP (multilayer perceptron)

predictions: 0.8, 0.5, 0.6, 0.7, 0.8
weights: 1, 2, 1, 2, 3
labels: 1, 0, 1, 1, 1
Sparse Linear Layer: Probability Calibration

MLP (multilayer perceptron) 

- Predictions: 0.8, 0.5, 0.6, 0.7, 0.8
- Weights: 1, 2, 1, 2, 3
- Labels: 1, 0, 1, 1, 1

- Sorted Predictions: 0.5, 0.6, 0.7, 0.8, 0.8
- Sorted Weights: 2, 1, 2, 3, 1
- Sorted Labels: 0, 1, 1, 1, 1
Sparse Linear Layer: Probability Calibration

MLP (multilayer perceptron)

- Predictions: 0.8, 0.5, 0.6, 0.7, 0.8
- Weights: 1, 2, 1, 2, 3
- Labels: 1, 0, 1, 1, 1

Sorted predictions: 0.5, 0.6, 0.7, 0.8, 0.8
Sorted weights: 2, 1, 2, 3, 1
Sorted labels: 0, 1, 1, 1, 1

Isotonic regression graph
Sparse Linear Layer: Probability Calibration

MLP (multilayer perceptron)
- Predictions: 0.8, 0.5, 0.6, 0.7, 0.8
- Weights: 1, 2, 1, 2, 3
- Labels: 1, 0, 1, 1, 1

Sorted predictions: 0.5, 0.6, 0.7, 0.8, 0.8
Sorted weights: 2, 1, 2, 3, 1
Sorted labels: 0, 1, 1, 1, 1

Binning
Sparse Linear Layer: Probability Calibration

The Model

MLP (multilayer perceptron)

- Predictions: 0.8, 0.5, 0.6, 0.7, 0.8
- Weights: 1, 2, 1, 2, 3
- Labels: 1, 0, 1, 1, 1

Sorted Predictions: 0.5, 0.6, 0.7, 0.8, 0.8
Sorted Weights: 2, 1, 2, 3, 1
Sorted Labels: 0, 1, 1, 1, 1

Compute average prediction per bin
Binning
Sparse Linear Layer: Probability Calibration: Closer Look

Isotonic regression

Data
Isotonic Fit
Linear Fit

Source: Scikit-learn
Timelines on Tensorflow
Timelines Model and Tensorflow Hub

MLP (multilayer perceptron)

Probability Calibration
Tensorflow: Profiler
Tensorflow: Tensorflow Model Analysis
Tensorflow: Tensorboard
Lessons Learned, Conclusions, Next Steps
Lessons Learned: Platform

- Unifying teams under a single unified platform
- ML Automation is crucial for productionizing models
- Visualization tools are really helpful for ML practitioners
Lessons Learned: Product

→ Choice of what to optimize for:
  ◆ Engagement? Time on app? Active days?
→ Feedback loop – training on previously served data
→ Automated and frequent model refreshes
→ Speed of development: Feature engineering, new modeling techniques
Conclusion

Goal of Timelines Product: Show users with most relevant Tweets

Huge scale

- Infrastructure constraints
- Modeling Decisions
- Tensorflow offerings used by Timelines
Next Steps

➤ Continuous Learning

➤ Model integration: automated model evaluation

➤ More model exploration for Timelines Quality in Tensorflow
Thank you!
Questions?

If you are interested in learning more about Twitter Cortex and Twitter Timelines Quality, please DM us on Twitter: @cibelemh @satanjeev