Embeddings @ Twitter
Making ML easy with Embeddings !!!
1 Team
2 What's an Embedding?
3 Why Embeddings?
4 Embeddings Pipeline
5 What’s Next
1 Team
2 What's an Embedding?
3 Why Embeddings?
4 Embeddings Pipeline
5 What's Next
To unify and advance recommendation systems.
Recommendation Systems
Notifications

News for you
LeBron James goes unfiltered in his upcoming barbershop-style show

Highlights from Aparna Nancherla and 12 others

News for you
Putin won't visit Washington before the Russia investigation ends, White House says

New Tweets from SurpriseSnacksNYC
Twitter
Agenda

1. Team and Product
2. What's an Embedding?
3. Why Embeddings?
4. Embeddings Pipeline
5. What's Next
What is an Embedding?

Discrete → Model → Continuous Space!

- twitter: [0.07, -0.001, -0.208]
- @jack: [0.427, 0.225, -0.082]
- SF: [0.541, 0.496, -0.362]
- #TwitterNBA: [0.414, 0.068, -0.196]
Two dimensional “user” embedding
1 Team and Product
2 What's an Embedding?
3 Why Embeddings?
4 Embeddings Pipeline
5 What’s Next
Why Embeddings?

**Model Features**
- Lead to improved model performance when used as input features

**Feature Compression**
- Reduced infrastructure cost and improved efficiency

**Nearest Neighbor Search**
- Similarity search on the embedding space

**Transfer Learning**
- Knowledge exchange between related domains while reducing training time and boosting performance
Model Features

- ML practitioners typically use one-hot encoding to represent categorical inputs
  - Incapable of encoding relationships
  - Sparsity issues make it less useful for large dimensions

- Embeddings are outputs of ML models
  - Conserve relationships amongst entities
  - Compress the sparse input space into dense vectors
Two dimensional “user” embedding
Model Features

Why Embeddings?
Why Embeddings?

Feature Compression
Feature Compression

Slow inference – lots of offline features
Why Embeddings?

- Generate embeddings from a sub-network offline
- Update at the same frequency as the raw features
Feature Compression

Serve online features + dynamic embedding
Why Embeddings?

Nearest Neighbor Search
Nearest Neighbor Search

Why Embeddings?
Nearest Neighbor Search

Why Embeddings?

- Essential component for Candidate Generation pipelines
  - Co-embed users and items
  - Given a user, lookup neighbors
  - Use approximate methods to scale

- Finds application in many other areas
Why Embeddings?

- Model trained for one task is used in another
  - Typically by initializing network weights and fine tuning
- Very attractive from a business point of view
  - Reduced development time
  - Cross domain information sharing
1. Team and Product
2. What's an Embedding?
3. Why Embeddings?
4. Embeddings Pipeline
5. What's Next
# Goals

<table>
<thead>
<tr>
<th>Quality and Relevance</th>
<th>Creation &amp; consumption with ease</th>
<th>Sharing &amp; discoverability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable adapting to evolving data distributions over time</td>
<td>Enable teams to learn embeddings at scale using the appropriate algorithm</td>
<td>Enable cross team collaboration</td>
</tr>
<tr>
<td>If applicable the learnt embeddings should be of value across product ML models</td>
<td>Enable teams to consume embeddings at scale</td>
<td>Improvements/learning in one domain can drive improvements elsewhere</td>
</tr>
</tbody>
</table>
Embedding pipeline

- ML workflow
  - Item selection
  - Data preprocessing
  - Model fitting
    - Benchmarking
  - Feature store
Item Selection & Data Preprocessing

- Identify the set of entities to learn embeddings for
- Assemble dataset that represents the relationships between these entities
  - Data representation defined by the learning algorithm
Model Fitting

- Fit a model on the collected data
  - Use pre-built algorithms
  - Option to plug in a custom algorithm
Benchmarking

- Developed a variety of standard benchmarking tasks for each type of embedding
Benchmarking

- Developed a variety of standard benchmarking tasks for each type of embedding
  - *User metadata prediction*: Predictive performance of a logistic regression model learnt on the users embedding.
Developed a variety of standard benchmarking tasks for each type of embedding

- **User Follow Jaccard:**
  Jaccard index of the users’ embedding similarity and their follow sets'
Feature Store

- Publish embeddings to the "feature store", Twitter's shared feature repository

- Enables ML teams throughout Twitter to easily discover, access, and utilize freshly trained embeddings.
  - Easy offline & online access
  - Discovery through UX
Agenda

1. Team and Product
2. What's an Embedding?
3. Why Embeddings?
4. Embeddings Pipeline
5. What's Next
Whats Next?

- New embedding learning algorithms
- Increasing number of datasets available as embeddings
- Large scale approximate nearest neighbor (ANN) solution
- Further exploration with embeddings as means for feature compression
Thank you

@tayal_abhishek

September, 2018
We are Hiring!!!
#TwitterCortex #MLX