What is Session-Based Recommendation?

- **Session-Based Recommendation (SBR):** Predict the next item with which a user will interact in a session.
- **E-commerce / bol.com:**
  - Given a sequence of items \( s = (s_1, s_2, ..., s_n) \), predict the next item \( s_{n+1} \).

Scalability Challenges in SBR

- Recommendations too costly to precompute due to the large number of potential sessions.
- SBR system needs to compute recommendations online and maintain state.
- Low latency response time (p90 < 50ms) required in real-world scenarios.
- High throughput (>1000 predictions/second).
- High-dimensional, extremely sparse click data from e-commerce platforms (33 million distinct items, very short sessions).

Efficient and High-Quality Recommendations

- Experimental study on bol.com data.
  - Replicated experimental study from Ludewig et al.: “Performance comparison of neural and non-neural approaches to session-based recommendation”, RecSys 19 on bol.com click data.
  - Vector-Session kNN (VS-kNN) approach outperformed neural approaches both in terms of prediction quality and inference time.
- Published as “Learnings from a retail recommendation system on Billions of interactions at bol.com” at ICDE’21.
- Design of the **VMIS-kNN** algorithm:
  - Adaptation of VS-kNN.
  - Leverages a precomputed index over historical click data for fast inference.
  - Minimises intermediate results during nearest neighbor search.
  - Highly tuned implementation in Rust.

Datasets for Offline Evaluation

- **Micro Benchmark for Inference**
  - **MicroBenchmark runtimes in microseconds (log-scale)** for VMIS-kNN vs. VS-kNN on the ecom-1m dataset with \( k=100 \).

Serenade

- **Offline computation of session index**
  - 180 days of click data.
  - (2.3 billion user-item interactions).
- **Data-parallel computation with pySpark**.
- **Online serving of next-item recommendations**
  - Replication of index (13 GB of memory required per machine).
  - Needs 2 CPU’s in total to handle 1000 req/s.
  - Colocation of evolving sessions with recommendation requests and session updates (using session affinity).
  - Maintenance of session state via a local key-value store (RocksDB).

Online A/B Test on the Live Platform

- **Experimental setup**
  - Serenade vs legacy system (item-to-item CF).
  - 3 week long A/B test for ‘other also viewed’ recommendations on the product detail page.
  - Training data: 562 millions user-item interactions after pruning, extracted from 180 days of data, 6.5M distinct items.
- **Results**
  - Test included 45 million user sessions.
  - Load varied between 200 and 600 requests per second.
  - Response latency: p90 around 5ms, p99.5 < 10ms.
  - 2.85% increase in relevant business metric.

Summary

- **Design and implementation of a kNN-based real-world SBR system**
- **Deployed in production at bol.com**
- Scalability achieved via:
  - Scalable variant of a well working KNN approach, based on a precomputed index for fast inference.
  - Colocation of sessions with recommendation requests on serving machines.
  - A/B test showed low-latency response times and increase in business metrics.
- **Systems publication:** Barrie Kersbergen, Olivier Sprangers, Sebastian Schelter. “Serenade - Low-Latency Session-Based Recommendation in e-commerce at Scale,” ACM SIGMOD, 2022 (to appear).
- More about our research on https://bkersbergen.github.io.

References