Database Systems Group • Prof. Dr. Thomas Seidl

Topics

Praktikum Big Data Science SS 2017
• Topics
  1. Subspace Clustering
  2. Search Engine
  3. Graph Learning
  4. Small Data

• Groups
In KDD1 and KDD2:

- learned several clustering models and algorithms
  - Density based, partitioning, hierarchical clustering
  - Subspace clustering (e.g. SUBCLU, CLIQUE)
  - Projected clustering (e.g. PROCLUS, PREDECON)
  - Correlation clustering (e.g. 4C, CASH)

In Big Data Management & Analytics:

- Learned about map-reduce
- Had map-reduce variant of k-means
**P3C^+ – MR**

- A projected/subspace clustering algorithm
- Suitable for large data sets in high-dimensional spaces
- Extends P3C by map-reduce
- Source:
  
  
  Projected Clustering for Huge Data Sets in MapReduce.

*International Conference on Extending Database Technology, 49–60.*
• **Primary objectives:**
  • Read and understand the $P3C^+ - MR$ paper and write a ‘documentation’ of how the algorithm works
  • Identify major steps/tasks of the algorithm
  • Implement the described map-reduce variant
  • Evaluate the algorithm
  • Create a UI in which the algorithm can be executed on input files (e.g. *.csv) and returns a visualization
Internet has a huge amount of text (and information)
How can we retrieve the information we are looking for?
⇒ Search Engine
Implement our own Search Engine using Apache Flink
• Implement a new search engine in a specific context
  • StackOverflow
  • Patent Dataset
  • Another dataset?
• Apply standard Information Retrieval algorithms (e.g. BM25 Score)

\[
BM25(d_j, q_{1:N}) = \sum_{i=1}^{N} IDF(q_i) \frac{TF(q_i, d_j)(k + 1)}{TF(q_i, d_j) + k \left(1 - b + b \frac{|d_j|}{L}\right)}
\]

• Use Information Extraction to find synonyms and improve the search engine
• Implement Question Answering (e.g. AskMSR)
• Search for the person who can be asked to answer this question, if no result satisfies the user
• Expected outcome:
  • Search algorithm (Okapi BM25) implemented in Flink
  • Query website
  • Information Retrieval/Extraction in Flink
  • Question answering
• Lots of interesting data has an intrinsic graph structure, e.g.
  • Social networks, sensor networks, citation networks, ...

• Typical graph learning tasks include
  • Node classification, link prediction, content recommendation, ...

• For these learning tasks, it is useful to first learn a latent vector space embedding of the nodes based on the graph structure
  • Learned node vectors can further be combined with other node features
• Deepwalk
  • Based on word embedding algorithm *word2vec* from NLP
    • Word representations are learned based on their context (Distributional Hypothesis - words in similar contexts are similar):

    ... how to stop *puppy* from *barking*...
    ... *barking dog* stole my sleep...

• Adaptation to learn graph node embeddings by sampling random walks to form „sentences“

```plaintext
B → A → C → D → E → D → C
```
• Goals
  • Get familiar with Flink’s graph API „Gelly“
  • Prepare the Deepwalk algorithm and related theory
  • Implement the Deepwalk algorithm in Apache Flink
  • Improve and optimize your implementation (and try different variations)
  • Evaluate your implementations
  • (Implement a stream version of the algorithm)
  • Think of an interesting use case
    • Apply your node embedding algorithm and solve a subsequent learning task on a real dataset (e.g. embedding of web graph and recommendation of similar websites)
    • Prepare a demo framework for your use case
• Resources
  • Papers
    • Intuition on word2vec: [https://deeplearning4j.org/word2vec](https://deeplearning4j.org/word2vec)
  • Datasets
    • [https://snap.stanford.edu/data/index.html](https://snap.stanford.edu/data/index.html)
    • [http://konect.uni-koblenz.de/](http://konect.uni-koblenz.de/)
• Why should we consider distributed computation for „small data“?
  • Dataset fits in one machine
  • Model can be learned in acceptable time on one core

• Find the best solution for the problem is tricky:
  • Different models (e.g. different classification algorithms)
  • Each model has different hyperparameters (grid search)
  • Cross-validation is often necessary for „small data“
  • Variance (e.g due to the random parameters initialization)

• Apply Map-Reduce to find the best model
• Solve real live problem: Predict traffic flow in small road network
  • Given current travel time, predict average travel time in one hour
  • Given current tollgate traffic volume, predict average traffic volume in one hour

• KDD Cup 2017 (last submission possibility June 1st)
• **Expected outcome:**
  • Selection of models for traffic flow prediction problem
  • Documentation of models and explanation of hyperparameters
  • Model selection framework in Flink
  • GUI for model selection framework for arbitrary dataset
  • Best model for traffic flow prediction problems