ViewSeeker: An Interactive View Recommendation Tool

Xiaozhong Zhang, Xiaoyu Ge, Panos K. Chrysanthis, Mohamed A. Sharaf
Table of Contents

• Background and Motivation
• Problem Definition
• The ViewSeeker
• Experimental Testbed and Results
• Conclusion
What is a View?

- A view is a histogram or bar chart showing the binned aggregated values over a dataset.
- A target view showing the player 3-point attempt rate of a NBA team.
An Insight from Views

- Create a reference view showing the attempt rate of all NBA players.
- The comparison between the two may explain why the selected team won a championship.

Views are powerful agents in helping the user explore and understand big data!
What is View Recommendation?

• Among all possible views, find the top-K views that are most useful according to the user’s interest.

• Existing view recommendation approaches proposed many view utility functions to estimate the usefulness of a view, e.g.
  • deviation
  • accuracy
  [Humaira Ehsan, et al. “MuVE: ... Visual Data Exploration.” In IEEE ICDE 2016.]
  • usability
  [Bo Tang, et al. “Extracting ... Multi-dimensional Data.” In ACM SIGMOD 2017.]
  • p-value
However...

There are too many view utility functions to choose from and we don’t know which one best captures the user’s interest.
Predefined View Utility Functions

- Recommendation precision of some predefined view utility functions:

Predefined view utility function cannot capture the user’s interest well.
Our Contribution

• The first attempt towards automatically discovering the most appropriate view utility function during *interactive view recommendation* that best captures the user’s interest.
Table of Contents

• Background and Motivation
• Problem Definition
• The **ViewSeeker**
• Experimental Testbed and Results
• Conclusion
Classical View Recommendation

• **INPUT**
  - A database $D$
  - A set of results $R$ produced by a user-specified query $Q$
  - A *predefined* view utility function $u()$

• **OUTPUT**
  - The top-$K$ views $V = \{v_1, v_2, ..., v_k\}$ constructed from $R$
  - The views in $V$ should have the highest utility among all possible views according to $u()$. 
Interactive View Recommendation

• INPUT
  • A database $D$
  • A set of results $R$ produced by a user-specified query $Q$
  • A set of $n$ possible utility functions $U = \{u_1, u_2, ..., u_n\}$
  • User feedback on example views
  • A time constraint $tl$
Interactive View Recommendation

• OUTPUT
  • A composite utility function $u^p()$, which can be any arbitrary combination of the utility functions in $U$.
  • The top-K views ranked by $u^p()$ should have high utility according to the user’s ideal utility function $u^*(())$.
  • The computational delay between each subsequent interactions (feedback) with the user is within $t_l$. 
Table of Contents

• Background and Motivation
• Problem Definition
• The ViewSeeker
• Experimental Testbed and Results
• Conclusion
ViewSeeker

Phase 1: Offline-Initialization

Stage 1: Calculate aggregated values for each view

Stage 2: Calculate utility features for each view

- KL-divergence
- EMD
- L1
- L2
- Max. Difference
- Usability
- Accuracy
- p-value
- ...

- Reference View
- Target View
ViewSeeker
Phase 2: Interactive View Recom.
ViewSeeker

Phase 2: Interactive View Recom.
ViewSeeker
Phase 2: Interactive View Recom.

Which view to select?  The **most informative** view!

Which view is the most informative?  The view whose utility the system is **most uncertain** about!
ViewSeeker Optimizations

Before interactive recommendation, calculate “rough” utility features using partial data
ViewSeeker Optimizations

During interactive recommendation, rank the views based on $u^p()$
ViewSeeker Optimizations

Refine utility features using all data for highly-ranked views
Table of Contents

- Background and Motivation
- Problem Definition
- The ViewSeeker
- Experimental Testbed and Results
- Conclusion
Performance Metric

• Top-k Precision:

\[
\text{Precision} = \frac{|V^* \cap V^p|}{k}
\]

• \(V^*\) is the top-k views recommended by \(u^*(\cdot)\)
• \(V^p\) is the top-k views recommended by \(u^p(\cdot)\)
Evaluation of Effectiveness

• Number of labeled views needed to reach a 100% precision

![Graph showing the number of labeled views needed for different Top K values.](image)
Evaluation of Effectiveness

- Number of labeled views needed to reach a 100% precision

On average, only need 9 – 16 labeled views

Two-Component $u^*()$  
Three-Component $u^*()$
Evaluation of Optimizations

• Utility Distance (UD):

\[ UD = \left( \frac{\sum_{v_i \in V^*} u^*(v_i) - \sum_{v_i \in V^p} u^*(v_i)}{k} \right) / k \]

• \( V^* \) is the top-k views recommended by \( u^*(\cdot) \)
• \( V^p \) is the top-k views recommended by \( u^p(\cdot) \)
• \( u^*(\cdot) \) is the ideal utility function
Evaluation of Optimizations

- Running time needed to reach UD = 0

![Bar chart showing comparison between No Opt. and With Opt. for different values of Top K and Three-Component u^*()](chart.png)
Evaluation of Optimizations

- Number of labeled views needed to reach UD = 0

- 43% reduction in running time
- 19% increase in user labeling effort
Conclusion

• **ViewSeeker** uses active-learning techniques to discover the most appropriate view utility functions during an exploration based on the user’s feedback.

• The optimization techniques significantly improve the runtime efficiency of the **ViewSeeker**.

• **ViewSeeker** outperforms baseline approaches by a significant margin for both synthetic and real-world datasets.
Thank You!

**ViewSeeker: An Interactive View Recommendation Tool**

Xiaozhong Zhang, Xiaoyu Ge, Panos K. Chrysanthis, Mohamed A. Sharaf

http://db.cs.pitt.edu

- This work was supported, in part, by NIH under award U01HL137159.