SkyRank: a unified toolkit for Skyline ranking, benchmarking and visualization

Mickaël Martin Nevot [1] **and Nael Turlure** [1]

1 Aix-Marseille Université, France

Summary

SkyRank is an open source Python framework designed to implement, compare and visualize various **Skyline (Börzsönyi et al., 2001) ranking algorithms implementations**, including dp-idp (Valkanas et al., 2014) improved with dominance hierarchy, RankSky, CoSky and DeepSky (Martin-Nevot & Lakhal, 2024). While these methods have been independently introduced in the literature, SkyRank provides a **unified, reproducible, and extensible platform** to evaluate and visualize them interactively.

SkyRank is designed for researchers, students, and practitioners working on **multi-criteria decision making**, **Pareto dominance**, and **database preference queries**. It includes a modular benchmarking backend, a LaTeX-compatible graph generator, and two graphical user interfaces (based on PyQt5 and Tkinter) for interactive data exploration.

Statement of need

While the Skyline operator is widely studied, **few open source tools** exist to explore, rank, and visualize Skyline points across various datasets and ranking models. Implementations are often scattered, hard to reproduce, and lack integrated visual support.

SkyRank addresses this gap by:

- Providing **ready-to-use implementations** of major Skyline ranking approaches.
- Offering **graphical interfaces** to run algorithms and compare results visually.
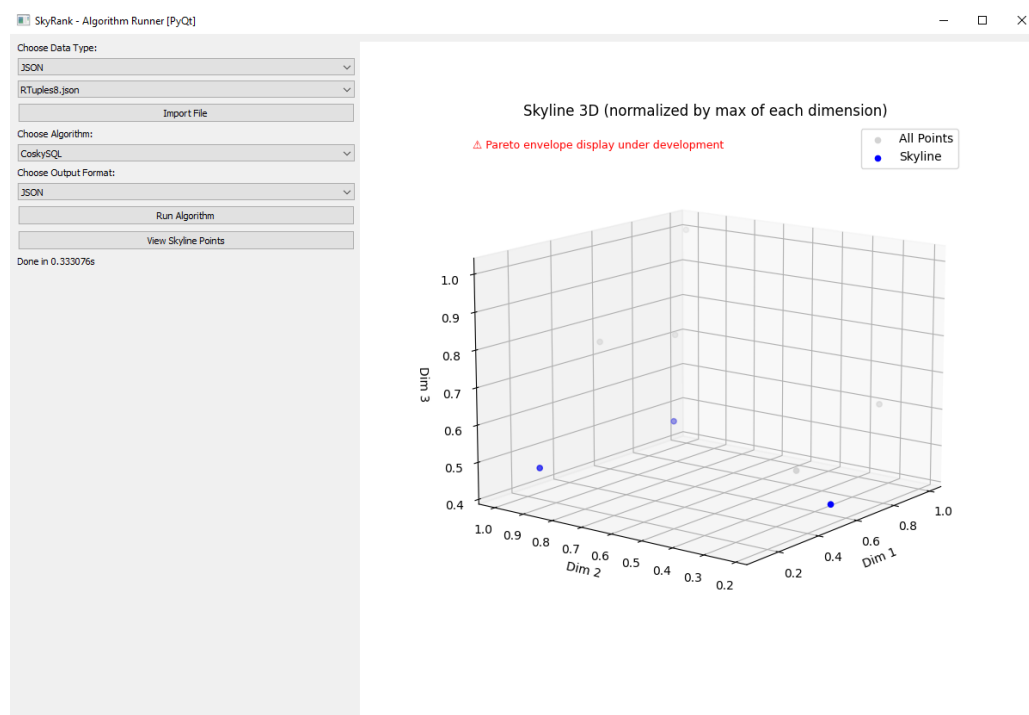
Figure 1: Graphical interface with Skyline points visualization.

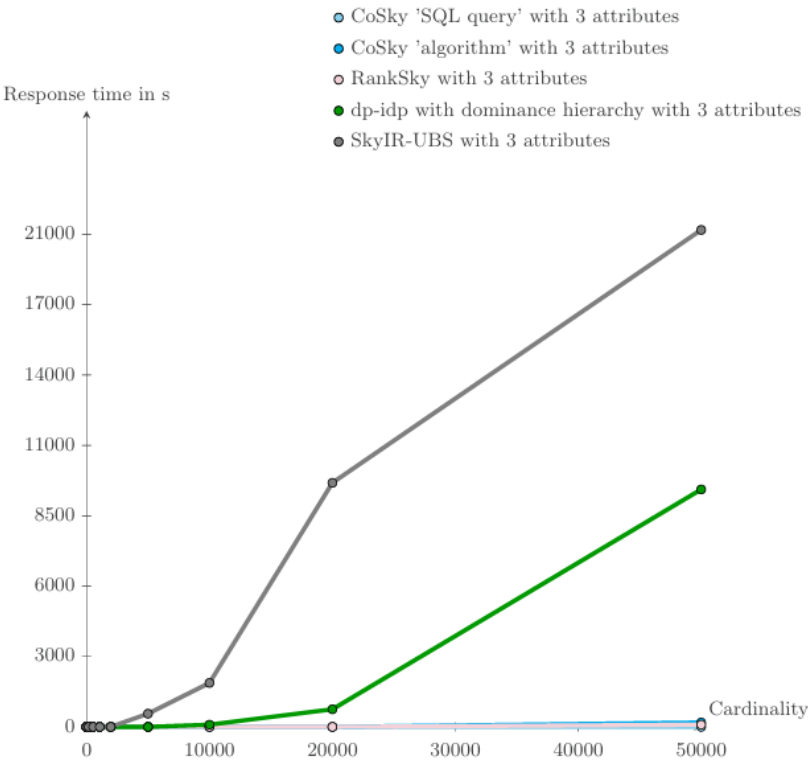- Supporting **LaTeX-based chart generation** for scientific reporting.

Figure 2: Latex graph generation.

24   ■ Enabling **fast experimentation** and educational use.

Features

26   ■   Implementations of `dp-idp` improved with dominance hierarchy, `RankSky`, `CoSky`,
27   `DeepSky`
28   ■   Automatic scoring using with dominance hierarchy, PageRank ([Page et al., 1998](#)) and
29   ([Langville & Meyer, 2006](#)), cosine similarity, and TOPSIS scoring ([Lai et al., 1994](#))
30   ■   Modular architecture for adding new ranking methods
31   ■   GUI interfaces for ranking and visual inspection of results
32   ■   Graph export in LaTeX/TikZ format
33   ■   Installation via script or editable pip package
34   ■   Benchmarking on synthetic uncorrelated datasets with 3, 6, and 9 dimensions, ranging
35   from 10 to 1,000,000,000 tuples

Software architecture

37   SkyRank relies on several Git repositories that are integrated as submodules to promote
38   modularity and reuse:

39   ■ [SkyRank-Client](#) serves as the main interface and execution environment for SkyRank,
40   allowing it to be used as a Python API.

- **BBS-Python-3.x-** provides an implementation of the BBS (Branch and Bound Skyline) algorithm.
- **R-Tree-Python-3.x-** implements spatial indexing structures required by BBS.

These repositories are hierarchically embedded as follows: `SkyRank-Client → SkyRank → BBS-Python-3.x- → R-Tree-Python-3.x-`.

This structure enables clean separation of features, encourages reusability, and simplifies algorithmic extensions across the Skyline ecosystem.

The repository is organized into logically separated modules, with each directory serving a dedicated purpose—from algorithmic implementation to UI, benchmarking tools, and export features—ensuring modularity and clarity throughout the codebase.

- `Algorithms/` contains the core implementations of Skyline ranking methods, including `dp-idp` improved with dominance hierarchy, RankSky, CoSky, and DeepSky.
- `Core/` hosts the application logic and UI entry points:
    - `App.py` handles global orchestration and job launching.
    - `AppUI.py` and `AppUIPyQT.py` provide Tkinter and PyQt5 graphical interfaces.
    - `LatexMain.py` generates ready-to-use LaTeX/TikZ charts.
- `Utils/` gathers reusable tools and utilities, organized by functionality:
    - `DataModifier/` and `DataTypes/` handle data preparation, data loading and internal structures.
    - `Exporter/` defines data export interfaces (e.g., CSV, LaTeX).
    - `Latex/` includes tools for LaTeX/TikZ generation.
- `Database/` provides database integration and mock data generation:
    - The `Database` class manages SQLite creation and insertion of random test data (with dynamic column scaling).
    - `SqlDataMocker` retrieves controlled subsets of rows/columns for benchmarking and converts them using a `DataParser`.
- `Assets/` contains benchmark datasets, test databases, execution logs, and configuration files.
- `docs/` includes documentation built with pdoc.
- `paper/` contains the JOSS submission material (`paper.md`, `paper.bib`).
- `external/` hosts Git submodules for third-party algorithms (BBS, RTree).

Acknowledgements

References

Börzsönyi, S., Kossmann, D., & Stocker, K. (2001). *The skyline operator*. 421–430. https://doi.org/10.1109/ICDE.2001.914855

Lai, Y. J., Liu, T. Y., & Hwang, C. L. (1994). TOPSIS for MODM. *European Journal of Operational Research*, *76*(3), 486–500. https://doi.org/10.1016/0377-2217(94)90282-8

Langville, A. N., & Meyer, C. D. (2006). *Google's PageRank and beyond: The science of search engine rankings*. Princeton University Press.

Martin-Nevot, M., & Lakhal, L. (2024). Classement d'objets skylines dans les bases de données. *arXiv Preprint arXiv:2411.02013*. https://arxiv.org/abs/2411.02013

Page, L., Brin, S., Motwani, R., & Winograd, T. (1998). *The PageRank citation ranking: Bringing order to the web*. Stanford Digital Libraries Technologies Project.

Valkanas, G., Papadopoulos, A. N., & Gunopulos, D. (2014). Skyline ranking à la IR. In K. S. Candan, S. Amer-Yahia, N. Schweikardt, V. Christophides, & V. Leroy (Eds.), *Proceedings of the workshops of the EDBT/ICDT 2014 joint conference (EDBT/ICDT 2014)* (Vol. 1133, pp. 182–187). CEUR-WS.org.