Quality-Performance Tradeoffs in Analytic and Machine Learning Workloads

Yongjoo Park (https://yongjoopark.com)

My research interest lies in building systems for interactive-speed data analytics and machine learning (ML). This area has been a focus of much academic research as well as industry efforts: Walmart has invested heavily in its cluster infrastructure, and last year alone, Ford invested $200M on its data center. However, the computational cost of large-scale data analytics and ML has remained daunting to small and large enterprises alike. Even major firms (e.g., Google, Ford, Walmart, Dunnhumby) constantly strive to reduce their infrastructure costs and query latencies. To meet these cost and latency goals, many data analysts and applications are *willing to tolerate a slight—but controlled—reduction of accuracy in exchange for substantial gains in cost and performance.* This is particularly true in early stages of data exploration, such as feature engineering, (hyper)parameter tuning, and visualization.

My research combines **rigorous statistical theories and large-scale data-intensive systems** to enable quality-performance tradeoffs in a principled manner. In particular, I build systems with three design goals in mind. First, the system must offer *high-level abstractions* that can effectively communicate these domain-specific tradeoffs to users who are not necessarily statisticians. Second, the system must exploit the unique data/workload characteristics to *infer* tight and reliable quality guarantees. Finally, this internal inference must be *computationally efficient* to maximize the performance gains of making these tradeoffs.

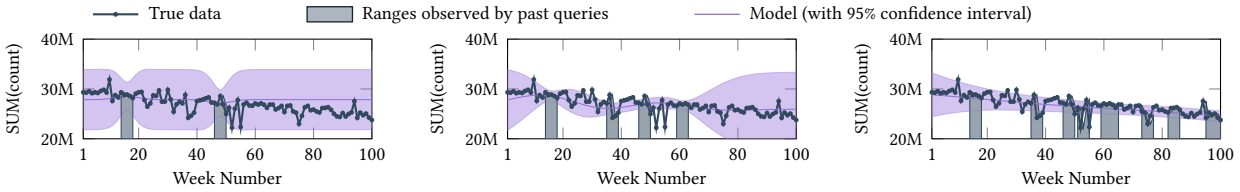
This approach has led to several publications (all first-authored) in premier database venues (e.g., SIGMOD, VLDB), a **2018 ACM SIGMOD Jim Gray Dissertation Runner-up Award**, and real-world adoption of my research results. In particular, one of my recently open-sourced systems, VerdictDB, has gained a growing userbase in the enterprise world, including **large-scale production deployments at Walmart [3], Digital2Go, and a major media company in Canada**, as well as test deployments at Alibaba and Tableau.

I have pursued my research agenda in several directions:

1. Quality-Performance Tradeoffs for **Autonomous Systems**: database learning [10], selectivity learning [11]
2. Quality-Performance Tradeoffs for **Exploratory Analytics**: SQL analytics [4, 8], visualization [7], image search [6]
3. Quality-Performance Tradeoffs for **Machine Learning**: maximum likelihood estimation [2, 9]

Quality-Performance Tradeoffs for Autonomous Systems

I have exploited quality-performance tradeoffs in building systems that become smarter over time (e.g., faster or more accurate) as they answer more queries. I have proposed *database learning* for approximate databases and *selectivity learning* for exact database systems.



(a) After 2 Queries (b) After 4 Queries (c) After 8 Queries

Figure 1: An example of database learning's probabilistic model construction.

Database Learning Database learning (DBL) [5, 10] is the first approximate query processing (AQP) system that can produce increasingly more accurate answers as it processes more queries. DBL builds a non-parametric probabilistic model of the underlying data by treating each answer as a piece of information about the underlying but unknown distribution of the data (see Figure 1). DBL then uses this *probabilistic model* to answer unseen queries faster and more accurately. In other words, instead of reading large volumes of raw data, we can use a small sample of it to quickly produce a sample-based approximate answer but then calibrate and combine this answer with the model to produce a more accurate answer to the query.

To realize DBL's vision in practice, we have overcome three key challenges. First, we have shown how to transform a wide class of SQL queries into appropriate mathematical representations so that they can be used by statistical methods for improving new queries. Second, to support arbitrary datasets, we have avoided any distributional assumptions about the underlying data. In other words, the only valid knowledge comes from past queries and their respective answers. Finally, we have stroke a balance between the computational complexity of our inference and its ability to reduce the error of query answers.

DBL covers 63.6% of TPC-H queries and 73.7% of a real-world query trace from a leading vendor of analytic DBMS. Formally, we also prove that DBL's expected errors are never larger than those of existing AQP techniques. In the experiments with both benchmark and real-world query traces, DBL delivers up to 23× speedup and 90% error reductions compared to regular AQP engines.

Selectivity Learning The selectivity estimation is a key step in nearly all cost-based query optimizers. Recently, we developed QuickSel [11], an ML-based selectivity estimation algorithm. QuickSel gradually refines its model of the underlying data by analyzing the selectivities of past queries. It then uses this model to estimate the selectivity of new, unseen queries. QuickSel makes a significant difference in practice: compared to the state-of-the-art adaptive histograms, QuickSel is 178×-313× faster; also, compared to periodically updated histograms and samples, QuickSel is 57.3% and 91.1% more accurate, respectively.

Quality-Performance Tradeoffs for Exploratory Analytics

I have developed systems and algorithms to enable interactive-speed exploratory analytics, expressed as SQL analytics, visualization, or search.

SQL Analytics Despite decades of research, *approximate query processing* (AQP)—trading off accuracy for faster query answers—has had little industry adoption. A few available AQP engines are each tied to a specific platform, and thus, require users to abandon their existing databases—an unrealistic assumption given the state of this industry.

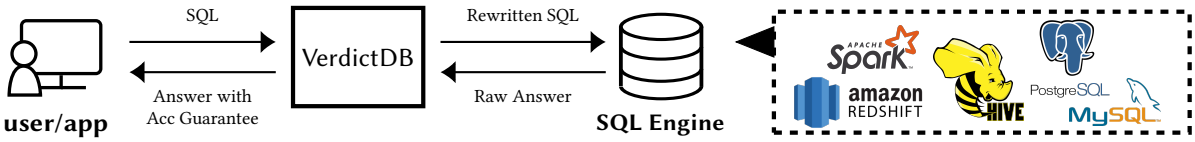


Figure 2: Architecture of VerdictDB. VerdictDB offers a speed-accuracy tradeoff on top of any existing SQL engine.

To close this long-standing gap between the AQP research in academia and industry, we have started an open-source project [1], called VerdictDB. VerdictDB introduces a radically different architecture: instead of modifying the query processing pipeline *inside* the database systems, VerdictDB operates at the driver-level, leaving the underlying database system intact. Specifically, VerdictDB is a thin layer between the user and the (off-the-shelf) SQL engine, which simply rewrites incoming queries, such that the standard execution of the rewritten queries under relational semantics would yield approximate answers to the original queries (see Figure 2). VerdictDB achieves a *platform-independent accuracy-guarantee* by relying on a new highly-efficient, SQL-based error-estimation technique, called *variational subsampling*, which yields provably-equivalent asymptotic properties to traditional subsampling. On industry-benchmarks, VerdictDB delivers 171× speedup (18.45× on average) across a variety of existing platforms, including Amazon Redshift, Apache Spark SQL, and Apache Impala, while incurring less than 2.6% relative error.

Visualization Interactive visualization is a key means of discovering trends in both science and business. However, visualizing a large dataset can be an extremely costly operation even with today's modern software. To tackle this challenge, we have devised a technique, called Visualization-Aware Sampling (VAS) [7], based on the following observation: including a data point in a plot matters only insomuch as it makes a visual difference. In other words, due to the finite number of pixels on a screen and the cognitive limitations of human vision in perceiving small details, we can characterize a notion of *visualization loss* and use that to produce an extremely small sample of the original dataset and yet enable high-fidelity visualizations. We have studied this problem in the contexts of regression, density estimation, and clustering tasks. The samples created by VAS speedup these visualization tasks by up to 400×.

Image Search NSH [6] is a hashcode-based k-nearest neighbor (kNN) algorithm that relies on a counter-intuitive idea. In general, the success of hashcode-based kNN techniques largely depends on their hash functions' ability to distinguish kNN items: that is, the kNN items retrieved based on data items' hashcodes, should include as many true kNN items as possible. A widely-adopted principle for this process is to ensure that similar items are assigned to the same hashcode so that the items with the hashcodes similar to a query's hashcode are likely to be true neighbors. NSH pursues a seemingly counter-intuitive (but provably more accurate) approach: it increases the distances between similar items in the hashcode space, instead of reducing it. This idea leads to 22.5% faster kNN compared to state-of-the-art methods.

Quality-Performance Tradeoffs for Machine Learning

Sampling is perhaps the most-widely used idea in practice for reducing the training time of ML models. However, most practitioners are not able to precisely capture the effect of sampling on the quality of their model. Most previous work on accuracy-guaranteed ML either targets a specific type of models (e.g., linear regression, k-means clustering) or requires data-sensitive pre-processing (e.g., coresets, QR factorization). As a more generic framework, we have developed BlinkML [9], a system that enables *fast* ML training with *probabilistic accuracy guarantees*. BlinkML supports any models that rely on maximum likelihood estimation for its training, including linear regression, logistic regression, max entropy classifier, and Probabilistic Principal Component Analysis. BlinkML's great efficiency stems from its novel mechanism that can estimate the accuracy of a sample-trained model *without actually training it*. BlinkML automatically determines the minimum sample size that can meet a user-requested accuracy guarantee, and assures that its models produce identical predictions as the full model (trained on the entire data) with high probability (e.g., 95%).

BlinkML therefore offers a flexible framework for making speed-accuracy tradeoffs. For example, it can train most models 6.26×–629× faster while guaranteeing the same predictions as the full model with 95% probability. Moreover, in feature engineering tasks [2], BlinkML can examine 320× more models compared to traditional approaches that train a full model.

Future Direction

As demonstrated in my research, building large-scale data-intensive systems that enable statistical tradeoffs is a powerful means of reducing computational costs and achieving interactive speeds at scale. However, we are still in an early stage of exploiting its full potential, and I see many more exciting opportunities in this area.

Framework for Short-lived, Stochastic Jobs An increasingly common but under-served class of tasks in a shared clusters are what I refer to as *short-lived* and *stochastic* jobs: they last less than a few minutes, and their execution paths may change dynamically depending on the distribution of data. Specifically, when quality-performance tradeoffs are exploited, *processing just a couple of million records can produce sufficiently accurate answers to many queries.* If intermediate answers are considered to be accurate enough, these jobs can finish early before processing all data. Moreover, since AQP and ML workloads typically consume (randomly) shuffled data, their intermediate results (i.e., aggregates in AQP or gradients in ML) are stochastic. To properly evaluate the quality of these stochastic results, the next generation of system runtimes must account for data-/workloads characteristics.

1. Distributed Engine for Stochastic Jobs: To best accommodate short-lived, stochastic jobs, the system must be able to process a large number of small batches of data with low-computational overhead and be able to offer a mechanism to constantly examine the accuracy of intermediate answers concurrently without blocking the main computations. The accuracy of these answers must be examined quickly on the fly; however, simultaneously, these accuracy examination must not much increase the overall processing time.
2. Extending Quality-Guaranteed Analytics: Unlike previous work that mostly focuses on traditional aggregations or specific ML models, BlinkML unleashes the power of uniform random sampling for a wide-class of ML models. I plan to extend this capability beyond that offered by VerdictDB or BlinkML. My first goal is offering quality-performance tradeoffs for more commonly used ML models (e.g., decision tree, Gaussian Process regression, support vector machines) and statistical tools (e.g., ANOVA, ARMA model, hypothesis testing). Then, I will further extend this capability to new, emerging models.

Autonomous Approximation-as-a-Service As more enterprises are moving their data analytics to the cloud, approximation-as-a-service presents itself as a unique opportunity. However, unlike on-premise deployments where

an experienced analyst can oversee the creation and maintenance of data synopsis, a hosted service requires autonomous operations. To realize this vision, I plan to work on Autonomous Sample Provisioning in my upcoming research. In particular, this new scheme will adaptively and gradually update samples and physical table partitions to effectively bound the query latencies for target workloads. Realizing this goal involves overcoming several key challenges, such as how to define the semantics of a target workload, how to unify sampling and partitioning strategies, how to automatically determine the appropriate update frequencies and enable rolling updates in this new context.

In summary, my five-year plan is to continue to widen the reach and impact of a new generation of data systems that enable quality-performance tradeoffs in a principled and user-friendly manner. My VerdictDB project will be an instrumental vehicle in creating an open-source community around this technology, where research problems are identified, and research solutions are developed and validated against real-world users. We already constantly receive feedback and feature requests from the community, which serves as a rich source of real-world insight into our research direction. I likewise plan to release additional layers on top of VerdictDB, including our recently proposed BlinkML framework.

References

[1] VerdictDB website. http://verdictdb.org/.

[2] M. R. Anderson, D. Antenucci, V. Bittorf, M. Burgess, M. J. Cafarella, A. Kumar, F. Niu, **Yongjoo Park**, C. Ré, and C. Zhang. Brainwash: A data system for feature engineering. *CIDR'13: The 8th biennial Conference on Innovative Data Systems Research*, Santa Cruz, USA, 2017. Vision.

[3] D. Goyal. Approximate query processing at WalmartLabs. https://fifthelephant.talkfunnel.com/2018/43-approximate-query-processing.

[4] W. He, **Yongjoo Park**, I. Hanafi, J. Yatvitskiy, and B. Mozafari. Demonstration of verdictdb, the platform-independent aqp system. *SIGMOD'18: ACM SIGMOD International Conference on the Management of Data*, Houston, USA, 2018. Demo.

[5] **Yongjoo Park**. Active database learning. *CIDR'17: The 8th biennial Conference on Innovative Data Systems Research*, Santa Cruz, USA, 2017. Abstract.

[6] **Yongjoo Park**, M. Cafarella, and B. Mozafari. Neighbor-sensitive hashing. *VLDB'16: 42nd International Conference on Very Large Data Bases*, New Delhi, India, 2016. Full Research.

[7] **Yongjoo Park**, M. Cafarella, and B. Mozafari. Visualization-aware sampling for very large databases. *ICDE'16: 32nd IEEE International Conference on Data Engineering*, Helsinki, Finland, 2016. Full Research.

[8] **Yongjoo Park**, B. Mozafari, J. Sorenson, and J. Wang. VerdictDB: universalizing approximate query processing. *SIGMOD'18: ACM SIGMOD International Conference on the Management of Data*, Houston, USA, 2018. Full Research.

[9] **Yongjoo Park**, J. Qing, X. Shen, and B. Mozafari. BlinkML: Efficient maximum likelihood estimation with probabilistic guarantees. *SIGMOD'19: ACM SIGMOD International Conference on the Management of Data*, Amsterdam, The Netherlands, 2019. Full Research.

[10] **Yongjoo Park**, A. S. Tajik, M. Cafarella, and B. Mozafari. Database Learning: Toward a database that becomes smarter every time. *SIGMOD'17: ACM SIGMOD International Conference on the Management of Data*, Chicago, USA, 2017. Full Research.

[11] **Yongjoo Park**, S. Zhong, and B. Mozafari. QuickSel: Quick selectivity learning with mixture models. *In Submission.* Full Research.