DOI:10.1145/3318219

To view the accompanying paper, visit doi.acm.org/10.1145/3318221 rh

# Technical Perspective
# Compressing Matrices for Large-Scale Machine Learning

By Zachary G. Ives

DEMAND FOR MORE powerful big data analytics solutions has spurred the development of novel programming models, abstractions, and platforms for next-generation systems. For these problems, a complete solution would address data wrangling and processing, and it would support analytics over data of any modality or scale. It would support a wide array of machine learning algorithms, but also provide primitives for building new ones. It would be customizable, scale to vast volumes of data, and map to modern multicore, GPU, coprocessor, and compute cluster hardware. In pursuit of these goals, novel techniques and solutions are being developed by machine learning researchers,[4,6,7] in the database and distributed systems research communities,[2,5,8] and by major players in industry.[1,3] These platforms provide higher-level abstractions for machine learning over data, and they perform optimizations for modern hardware.

Elgohary et al.'s work on "Scaling Machine Learning via Compressed Linear Algebra," which first appeared in the *Proceedings of the VLDB Endowment*,[2] seeks to address many of these challenges by applying database ideas (cost estimation, query optimization, cost-based data placement and layout). It was conducted within IBM and Apache's SystemML declarative machine learning project. The paper shows just how effective such database techniques can be in a machine learning setting. The authors observe that the core data objects in machine learning (feature matrices, weight vectors) tend to have regular structure and repeated values. Machine learning tasks over such data are composed from lower-level linear algebra operations. Such operations generally involve repeated floating-point computations, which are bandwidth-limited as the CPU traverses large matrices in RAM.

The authors developed a compressed representation for matrices, as well as compressed linear algebra operations that work directly over the compressed matrix data. Together, these reduce the bandwidth required to perform the computations, thus speeding performance dramatically. The paper cleverly leverages ideas from relational database systems: column-oriented compression, sampling-based cost estimation, and trading between compression speed and compression rate.

This paper makes several notable contributions. First, the authors identify a set of linear algebra primitives shared by multiple distributed machine learning platforms and algorithms. Second, they develop compression techniques not only for single columns in a matrix, but also "column grouping" techniques that capitalize on correlations between columns. They show that offset lists and run-length encoding offer a good set of trade-offs between efficiency and performance. Third, the paper develops cache-conscious algorithms for matrix multiplication and other operations. Finally, the paper shows how to estimate the sizes of compressed matrices and to choose an effective compression strategy. Together, these techniques illustrate how database systems concepts can be adapted to great benefit in the machine learning space. C

> **The authors developed a compressed representation for matrices, as well as compressed linear algebra operations that work directly over the compressed matrix data.**

**References**

1. Abadi, M. et al. Tensorflow: A system for large-scale machine learning. *OSDI, 16* (2016), 265–283.
2. Ewen, S., Tzoumas, K., Kaufmann, M. and Markl, V. Spinning fast iterative data flows. In *Proceedings of VLDB Endow. 5*, 11 (2012), 1268–1279.
3. Ghoting, A. et al. SystemML: Declarative machine learning on MapReduce. *ICDE*. IEEE, 2011, 231–242.
4. Low, Y. et al. GraphLab: A new parallel framework for machine learning. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*. (Catalina Island, CA, July 2010).
5. Meng, X. et al. Mllib: Machine learning in Apache Spark. *JMLR, 17*, 1 (2016), 1235–1241.
6. Paszke, A. et al. Automatic differentiation. PyTorch, 2017.
7. Team, T.T.D. et al. Theano: A python framework for fast computation of mathematical expressions. arXiv preprint arXiv:1605.02688, 2016.
8. Zaharia, M., Chodhury, M., Franklin, M.J., Shenker, A. and Stoica, I. Spark: Cluster computing with working sets. *HotCloud* 10, 2010.

**Zachary G. Ives** is Department Chair and Adani President's Distinguished Professor of computer and information science at the University of Pennsylvania, Philadelphia, PA, USA. He is also a co-founder of Blackfynn, Inc., a company focused on enabling life sciences research and discovery through data integration.