Data Expressiveness and Its Use in Data-centric AI

H. Kurban, P. Sharma,* M.M. Dalkilic
Computer Science Department
Indiana University, Bloomington, IN, USA

Abstract

To deal with the unimaginable continual growth of data and the focus on its use rather than its governance, the value of data has begun to deteriorate seen in lack of reproducibility, validity, provenance, *etc.* In this work, we aim to simply understand what is the value of data and how this basic understanding might affect existing AI algorithms, in particular, EM-T (traditional expectation maximization) used in soft clustering and EM$*$ (a data-centric extension of EM-T). We have discovered that the value of data–or its "expressiveness" as we call it–is procedurally determined and runs the gamut from low expressiveness (LE) to high expressiveness (HE), the former not affecting the objective function much, while the latter a great deal. By using balanced binary search trees (BST) (complete orders) introduced here, we have improved on our earlier work that utilized heaps (partial orders) to separate LE from HE data. EM-DC (expectation maximization-data centric) significantly improve the performance of EM-T on big data. EM-DC is an improvement over EM$*$ by allowing more efficient identification of LE/HE data and its placement in the BST. Outcomes of this, aside from significant reduction in run-time over EM$*$, while maintaining EM-T accuracy, include being able to isolate noisy data, convergence on data structures (using Hamming distance) rather than real-values, and the ability for the user to dictate the relative mixture of LE/HE acceptable for the run. The Python code and links to the data sets are provided in the paper. We additionally have released an R version (`https://cran.r-project.org/web/packages/DCEM/index.html`) that includes EM-T, EM$*$, and k++ initialization.

1 Introduction

More than 50 years ago the well-known physicist Feynman observed what he believed to be a looming problem which we paraphrase here: as computing technology advances, the ratio of time devoted to computing over the data to the time required to move data will tend toward zero [1]. We are now routinely given summary statistics that point to an encroaching 100 zettabytes of data–growing exponentially within this decade–while computing power is, at best, growing linearly. Exacerbating this problem is the almost uniform urgency to enhance model performance–leaving invaluable properties of data neglected *e.g.*, annotation, provenance, validity, maintainability, accessibility, security, reproducibility. We believe a culture of *data indifference* has arisen that treats data as nothing more than a kind of fuel (for modelling) whose governance, if at all, is relegated to making repositories whose support is typically erratic and short-lived. Feynman's observation will likely hold for the foreseeable future, and while his prescient observation was unlikely addressing data-centric AI directly, it has inspired our work to pose fundamental questions about data–what is its value? can it be determined? how this notion might yield insight into improving long-standing AI algorithms and heuristics.

The contribution of the work presented here is to: (1) refine our approach to determining value of data which we call *data expressiveness* whose semantics is determined procedurally (2) apply

*Email: parishar@iu.edu

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

this characterization using balanced binary search trees (BST) to improve a popular AI algorithm - expectation maximization (EM) which has had an earlier data-centric improvement using heaps called EM$*$. We call this improved data-centric version EM-DC (EM with data-centrism).

Not only are we able to significantly improve the run-time across feature size, data size, number of clusters, but we have as a direct result a means of establishing "good" from "bad" **(informally the learning agent is familiar (or recalls) good, but not bad)** allowing, for instance, more effective curation, a new characterization of noise, and a novel data-centric convergence on a *data structure* rather than a real-valued function limit. The remainder of the paper gives two introductory examples of data expressiveness and its use in learning; abridged background and related work; a discussion of expressive data structures–heaps and balanced binary search trees; methods in our experiments for EM-DC; experiments; conclusion and summary.

1.1 Learning with Data Expressiveness: Two Small Examples

The foundation of this work is understanding the value of data. We have argued that the notion of data value cannot be statically determined but is determined procedurally through interaction with computation over the life of the computation. We present here a simple example juxtaposing a simply learning problem that is not data-centric to one that is illustrating the potential benefit of separating good from bad data. Suppose we have a small deck of vocabulary card to help us learn a foreign language. Each card as a word on the front and definition on the back, *e.g.*, **front**: `arcānus,-a,-um` and **back**: `closed, secret (1st, 2nd declension) cognate: arcane`. The learning task is to be able to define all the words in the deck. Let $Error : D \rightarrow \{0, 1\}$ return 1 if an error is made, zero otherwise. A general iterative optimization learning algorithm can be designed and implemented:

`Visit` $d \in D$ `until` $\sum_{d \in D} Error(D) = 0$. This is not data-centric AI. Alternatively, we allow two structures to exist: one for high expressive data (HE)–words that are not known or "bad" data; one for low expressive data (LE)–words that are known or "good" data. HE substantially affects the error function (in this case, as the value one). LE, on the other hand, does not. There is a gamut between the two, but for now we assume a crisp valuation. Say we know three of the words from their cognates. After viewing them once, they can be added to the LE structure. Perhaps, after a couple of runs, we recall `bis` as `twice` remembering its use in a chemistry class. We are migrating the HE data to the LE structure. The convergence criteria is now on the empty HE structure. While in the worst case the data-centric run-time will be equal to non-data-centric run-time, in practice it will perform much better. As a second example, given in Fig. 1, we compare the number of iterations between traditional EM (EM-T) and data-centric EM (EM$*$) over a tiny data set of three pairs of points $D = [(1, 1), (2, 2), (5, 1)$. The run-time is reduced by half while yielding comparable accuracy.

2 Abbreviated Background and Related Work

Space prohibits giving a complete background and related work, but we point to [2, 3, 4, 5] for a thorough treatment. There has been scant work and less success in making EM-T work on big data [6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]. Our work allows for EM$*$ to work on big data.

2.1 Expressive Data Structures: Heaps and Binary Search Trees

In our earlier work a non-linear hierarchical data structure (max-heap) is embedded into EM-T that allows (1) separation of LE/HE data so that that HE is used rather than LE/HE entirely. We discovered a useful, novel property of heaps (that we are currently studying as well) that allows convergence what we call *strong* and *weak* forms that yield, respectively, LE and HE data that exhibit (or violate) a combinatoric property[17]. As heaps are built, tested, and torn down, LE data flows to the top and HE to the leaves. As iteration continues, we can alternatively ignore the LE portions of the heap during the expectation phase. What is fascinating is that as the algorithm iterates, the proportion of strong heaps grow, essentially monotonically, leaving only HE in the leaves indiscriminate in their favoring *any* heap. The correspondence between heap location and data expression is natural: strong heaps (from root to down) contain LE that, on insertion, do not change the existing LE positions. Weak heaps (from leaves to root) contain HE and will affect the heap structure greatly. Strong heaps can permute successive levels from the root and maintain the heap property; weak heaps cannot. We have conducted many experiments confirming that LE data migrates to the top of the heaps

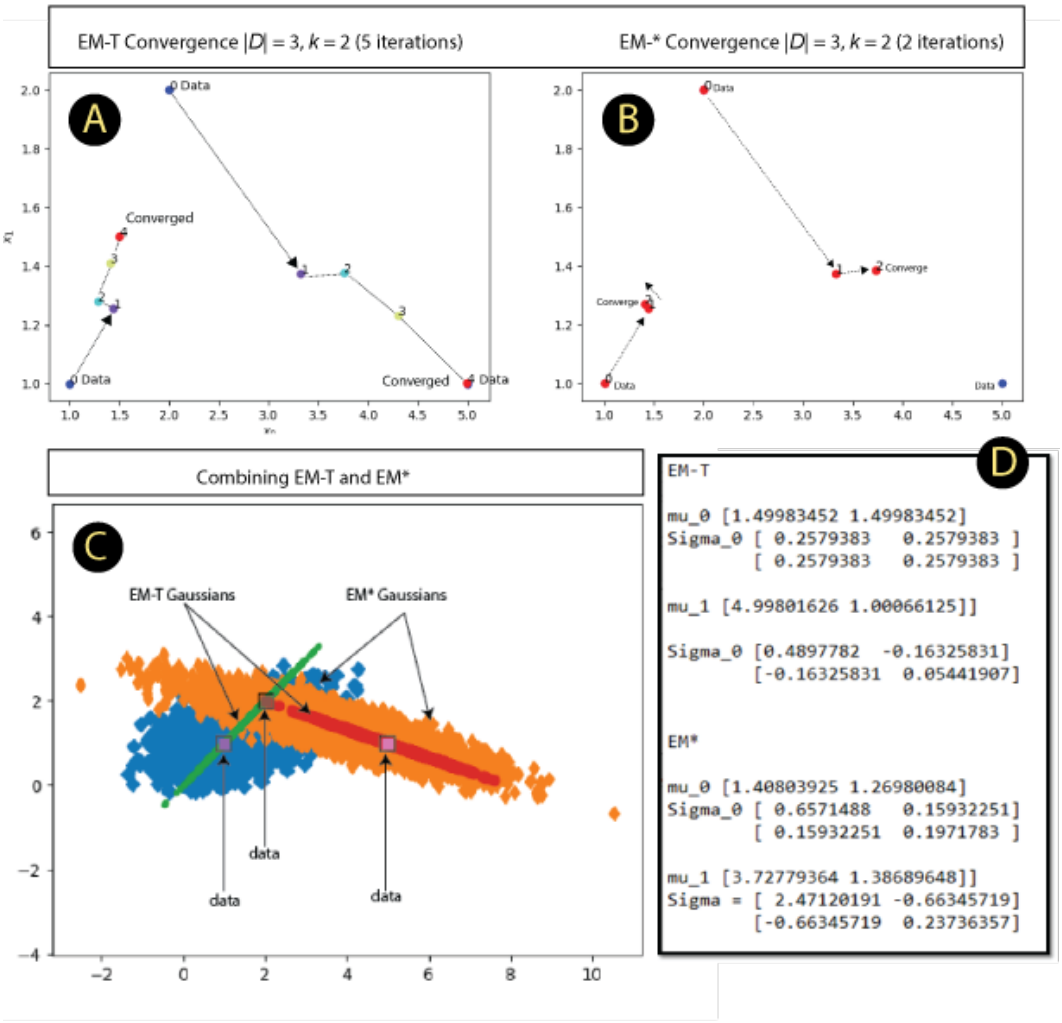

Figure 1: (A) EM-T *versus* (B) data-centric EM∗ with a small data set. Each arrow is a step. The iterations for EM∗ are about 1/2 of EM-T. (C) is a overlay of both showing the variance, data points, and paths toward convergence. (D) shows actual μ, Σ with μ between EM-T and EM∗ being unexpectedly close with disparate iterations.

while HE moves to the leaves. When the *Hamming distance* of the heaps does not change, we have converged. The leaf data can certainly be viewed as noise or, even worse, as "bad" data, but it is dependent on the corpus and hyperparameters chosen. In the work presented here, we have replaced heaps with binary search trees (total orders). This allows faster, more accurate determination of locations of expressiveness. In this case, HE data will be inserted into the leftmost branch always. Convergence, while determined by Hamming distance, is mirrored by the decreasing overall count of tree re-balancing (as LE flows the right and down and HE flows to the left and down). What remains on the left side as we move toward the leaves is HE data. We need to address limitations of this approach. Since we define value procedurally, a single D can take on multiple values using a single algorithm with randomness. One can easily imagine running the same algorithm, but producing different values for the same data. Provenance, comparison, maintainability, correctness, for example, now have heavier burdens to bear. Further, non-iterative techniques will not likely work with data expressiveness.

3 Experiments (Brief Overview)

The experimental results highlight the performance gain when comparing EM-DC with EM∗ and EM-T. While we performed an exhaustive set of experiments, due to space constraints, we only

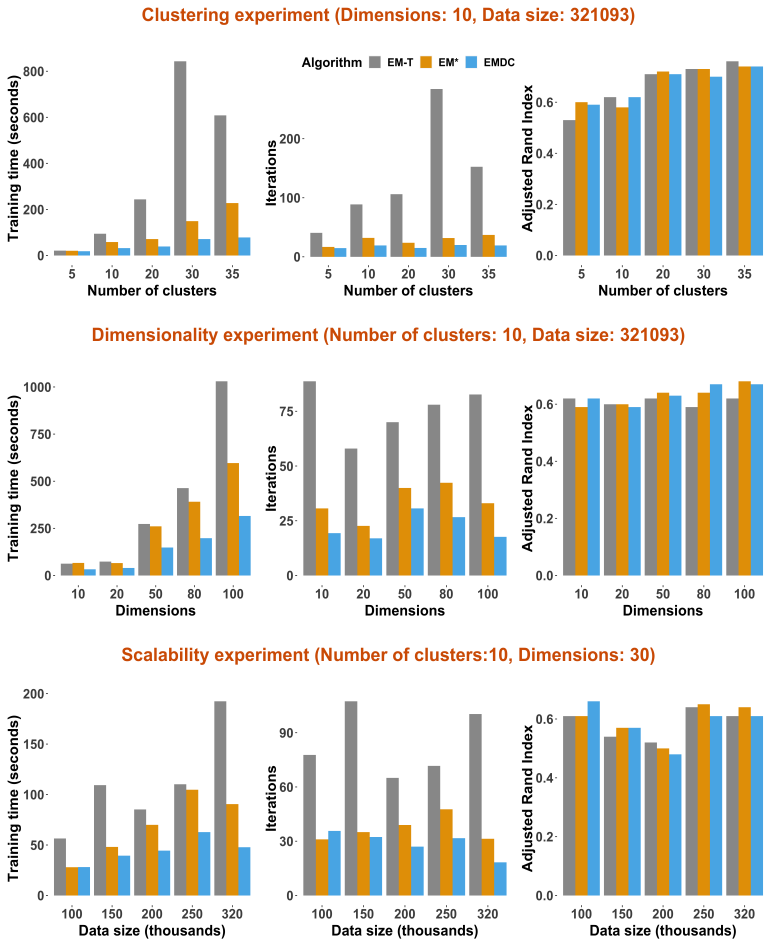

Figure 2: Comparison of EM *vs.* data-centric EM on real-world data across different aspects. Plots underscores that EM-DC is noticeably faster than EM-T and EM$*$ regardless of the number of clusters, feature size or data size. In general, data-centric AI outperforms its non-data-centric counterpart.

show the results from the cropland classification data set [18] (there is no personally identifiable information or offensive content). The data has 325,834 records in 175 attributes and contains 7 classes. To balance the class distribution, we only consider the 5 dominant classes that account for 97% of data. Experimental findings show that EM-DC is as good or significantly better than both EM-T and EM$*$ (Figure 2) and, the performance gap widens with increment in the number of clusters, dimensions and data size. For training time and number of iterations, EM-DC significantly outperforms both EM-T and EM$*$. In terms of classification accuracy (adjusted rand index), all algorithms perform similar. It should be noted that considerable reduction in total training time and number of iterations is achieved by EM-DC while using only a third of the data in the structure (*i.e.,* balanced binary search tree) whereas, EM$*$ use data in all leaf nodes and EM-T uses all the data. More details on how to reproduce the results, experimental setup, data, and parameter settings can be obtained from `https://github.com/parichit/EM-DC-NEURIPS_2021`.

4 Summary & Conclusion

Data-centric AI should include a fundamental understanding of the value of data. We propose expressiveness as a function of the coupling of data with an algorithm. We show that hierarchical structures can capture expressiveness and leverage this valuation to improve some long-standing AI techniques. In this work we show that a data-centric EM handily outperforms a non-data-centric EM.

References

[1] R.P. Feynman. *Lectures on Computation (Frontiers in Physics) 1st Ed.* CRC Press, 2000.

[2] H. Kurban. A Novel Approach to Optimization of Iterative Machine Learning Algorithms Over Heap Structures (Thesis), 2017.

[3] H. Kurban, M. Jenne, and M.M. Dalkilic. Using Data to Build a Better EM: EM* for Big Data. *International Journal of Data Science and Analytics*, 4(2):83–97, 2017.

[4] Michiko Watanabe and Kazunori Yamaguchi. *The EM algorithm and related statistical models*. CRC Press, 2003.

[5] Xiao-Li Meng and Donald B Rubin. Maximum likelihood estimation via the ecm algorithm: A general framework. *Biometrika*, 80(2):267–278, 1993.

[6] Paul S Bradley, Usama Fayyad, Cory Reina, et al. Scaling em (expectation-maximization) clustering to large databases. *Microsoft Research*, pages 0–25, 1998.

[7] Fredrik Farnstrom, James Lewis, and Charles Elkan. Scalability for clustering algorithms revisited. *ACM SIGKDD Explorations Newsletter*, 2(1):51–57, 2000.

[8] Carlos Ordonez and Edward Omiecinski. Frem: fast and robust em clustering for large data sets. pages 590–599, 2002.

[9] Kenneth Lange. A quasi-newton acceleration of the em algorithm. *Statistica sinica*, pages 1–18, 1995.

[10] Thomas A Louis. Finding the observed information matrix when using the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 44(2):226–233, 1982.

[11] Mary J Lindstrom and Douglas M Bates. Newton—raphson and em algorithms for linear mixed-effects models for repeated-measures data. *Journal of the American Statistical Association*, 83 (404):1014–1022, 1988.

[12] Geoffrey J McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions*, volume 382. John Wiley & Sons, 2007.

[13] Mortaza Jamshidian and Robert I Jennrich. Conjugate gradient acceleration of the em algorithm. *Journal of the American Statistical Association*, 88(421):221–228, 1993.

[14] Mortaza Jamshidian and Robert I Jennrich. Acceleration of the em algorithm by using quasi-newton methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(3):569–587, 1997.

[15] Martin A Tanner and Wing Hung Wong. The calculation of posterior distributions by data augmentation. *Journal of the American statistical Association*, 82(398):528–540, 1987.

[16] James F Epperson. *An introduction to numerical methods and analysis*. John Wiley & Sons, 2021.

[17] Hasan Kurban and Mehmet M Dalkilic. A novel approach to optimization of iterative machine learning algorithms: over heap structure. *IEEE*, pages 102–109, 2017.

[18] Iman Khosravi and Seyed Kazem Alavipanah. A random forest-based framework for crop mapping using temporal, spectral, textural and polarimetric observations. *International Journal of Remote Sensing*, 40(18):7221–7251, 2019. doi: 10.1080/01431161.2019.1601285.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] Sec. 1, Sec. 2.1, Sec 4
   (b) Did you describe the limitations of your work? [Yes] Sec. 2.1
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A] Theoretical elements are in the primary work cited with assumptions.
   (b) Did you include complete proofs of all theoretical results? [N/A] Proofs to theories, lemmas, *etc.*, are in primary work cited.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Sec. 3 through github
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Sec. 3 through github
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] 3 through github
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Sec. 3 through github
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Sec. 3
   (b) Did you mention the license of the assets? [Yes] Sec. 3
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Sec. 3 through github
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] There are not any.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] Sec. 3
5. If you used crowd sourcing or conducted research with human subjects
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] There was none.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] There was none.
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]