Education, Learning and Information Theory

Bryan Hooi*, Hyun Ah Song*, Evangelos Papalexakis*, Rakesh Agrawal† and Christos Faloutsos*
*Carnegie Mellon University
Email: bhooi@andrew.cmu.edu, hyunahs@cs.cmu.edu, epapalex@cs.cmu.edu, christos@cs.cmu.edu
† Data Insights Laboratories
Email: ragrawal@acm.org

***Abstract*—Suppose you are a teacher, and have to convey a set of object-property pairs ('lions eat meat'; or 'aspirin is a blood-thinner'). A good teacher will convey a lot of information, with little effort on the student side. Specifically, given a list of objects (like animals or medical drugs) and their associated properties, what is the best and most intuitive way to convey this information to the student, without the student being overwhelmed? A related, harder problem is: how can we assign a numerical score to each lesson plan (i.e. way of conveying information)? Here, we give a formal definition of this problem of forming learning units and we provide a metric for comparing different approaches based on information theory. We also design a multi-pronged algorithm, HYTRA, for this problem.**

Our proposed HYTRA is *scalable* (near-linear in the dataset size); it is *effective*, achieving excellent results on real data, both with respect to our proposed metric, but also with respect to encoding length; and it is *intuitive*, conforming to well-known educational principles, such as grouping related concepts, and "comparing" and "contrasting". Experiments on real and synthetic datasets demonstrate the effectiveness of HYTRA.

I. Introduction

There is great interest in developing tools that help deliver high quality, personalized education at large scale. To construct such systems, we need objective criteria for what a desirable curriculum is. One such criterion is that a good curriculum minimizes the complexity of the information to be conveyed: [1] conducted a series of experiments which support the hypothesis that the subjective difficulty of a concept is directly proportional to its Boolean complexity.

In our formulation, our facts consist of simple *(object, property)* pairs: for example, the object 'tiger' has the property 'striped.' Informally, our problem can be stated as follows:

*Informal Problem 1 (*Transmission Rate Problem*):* Given a large, sparse binary matrix whose rows represent objects, columns represent properties, in which ones represent facts, how do we measure how good a particular encoding of the matrix is for student learning, and how do we optimize this metric?

By constructing a binary matrix of facts, the *Transmission Rate Problem* becomes that of optimally encoding the information in a sparse binary matrix. However, since our goals are educational, our problem differs from the classical sparse matrix compression problem in that we seek *consistent learning* and not maximal compression. That is, we prioritize schemes which send some information as early as possible, thus giving learners gradual progress, over revealing information in one huge chunk. For example, a course where students can only decode the entire course material in the final lecture would not be ideal. As such, rather than using total encoding length as our metric, we formulate alternate metrics which prioritize consistent learning.

Reproducibility: All datasets we use are publicly available, and we make our implementation of HYTRA available for download at http://www.cs.cmu.edu/~hyunahs/tol, promoting the reproducibility of our results and facilitating future research on the subject.

II. Proposed Metric

Unlike compression length, our metric prioritizes teaching incrementally - that is, conveying useful chunks of information as early as possible. Our metric has a natural and intuitive interpretation as minimizing student effort, as well as maximizing students' utility. Our goal is to teach a collection of facts while minimizing student effort. We model student effort by the number of bits transmitted, since it represents the amount of attention that students need to understand the lesson content. As such, *ALOC* uses the number of bits transmitted as its measure of student effort (i.e. as the units along the x-axis when plotting $f(n)$).

Definition 1 *Area Left of Curve* (*ALOC*) Given an encoding algorithm, the *ALOC* metric is the area left of the curve $f(n)$, where $f(n)$ is the number of nonzero entries of the matrix that are decodable based on the first n bits output by the encoding algorithm. Lower *ALOC* is better.

As a simple example, assume that our data is the matrix $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, and we encode the upper-left one as $(0,0)$ in binary (00), then the lower-right one as $(1,1)$ in binary (11), resulting in the encoding 0011. Then the first one is decodable after 2 bits and the second after 4 bits, i.e. $f(0) = 0, f(1) = 0, f(2) = 1, f(3) = 1, f(4) = 2$, and the *ALOC* is 6.

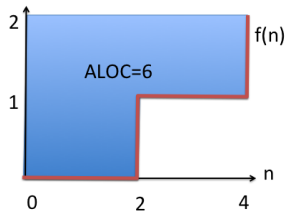

Fig. 1: f curve and ALOC for simple example.

III. Proposed Method (HYTRA)

According to the correspondence between encoding schemes and probability distributions [2], it is highly unlikely

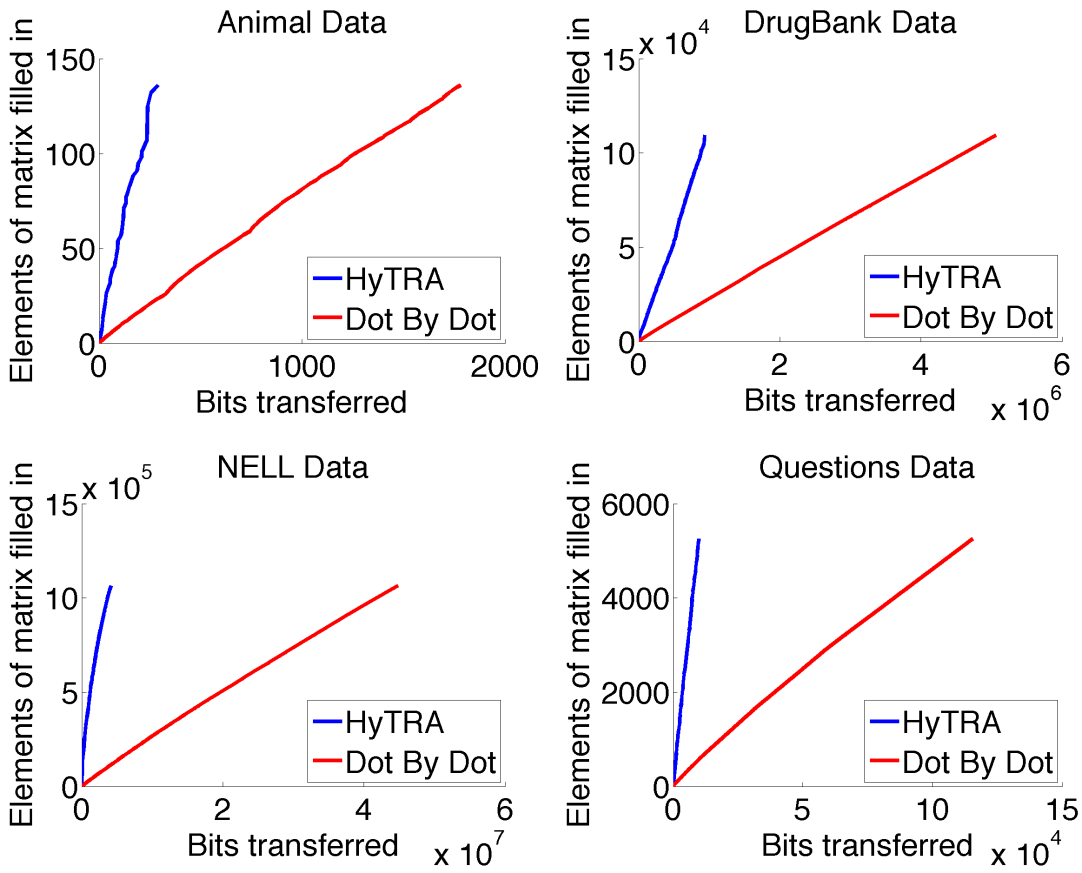

Fig. 2: HYTRA encodes data more efficiently than the baseline, both under total encoding length and *ALOC*. Curves with lower area to their left are better (under *ALOC*), and the total encoding length is the distance between the y-axis and the top of the curve.

for a single encoding scheme to perform exceptionally well on all datasets. This motivates a **multi-pronged approach**: we propose 4 sub-methods, each designed to perform well on a particular type of dataset. Our multi-pronged algorithm, HYTRA, tries each sub-method to encode a dataset, and chooses the encoding with lowest *ALOC*. This allows HYTRA to do well in diverse settings. HYTRA-Fishbone is designed to work well on data with highly skewed row or column sums such as hyperbolic structures, and works by rearranging the data to create a dense region at the upper-left of the matrix. Both HYTRA-Chain and HYTRA-Tree work well on data with similar rows or columns, and work by encoding a row based on comparison to the most similar other row. HYTRA-Onion works well on block structured datasets, and works by encoding blocks of ones. (full details of our algorithms can be found in our technical report at http://www.andrew.cmu.edu/user/bhooi/hytra.pdf) We add a pre-amble (the index of the chosen sub-method in binary) to the start of the encoding to indicate which sub-method was used, to ensure unique decodability.

A. Proposed encoding structures

Table I summarizes the encoding structure and keywords each sub-method uses to encode information, i.e. its encoding language. *r-id* and *c-id* refer to *row index* and *column index*, respectively. These encoding structures allow HYTRA to agree with the instructional principles of linking (linking related concepts), pre-conceptions (using existing knowledge to teach new facts) and comparison (comparing similarities and differences).

Under Onion, each statement in the encoding describes a block. It does this by encoding the block's row indices, column indices, and the missing positions in each block (**is-missing**). Both Tree and Chain using the same encoding structure: each statement encodes a row by comparing it to another row. For example, we may encode row representing tigers by the statement 'tigers are like lions, except they have stripes,' which would be encoded in the format 'row i is like row j except at columns k.' Note that Tree and Chain use different algorithms for choosing which other row to use to encode each row, which we detail in our full report. Under Fishbone, each statement describes a dense consecutive series of ones (i.e. a 1 by n block) by encoding its length n, then encoding each position of a missing one in the block.

IV. EXPERIMENTS

In this section we demonstrate the efficiency and effectiveness of HYTRA using real and synthetic datasets. We

Method	Encoding structure	Keywords	Encoding
Onion	*(r-id)* + **Row-like** + *(r-id)* *(c-id)* + **Column-like** + *(c-id)* *(r-id)* + **is-missing** + *(c-id)*	**Row-like** **Column-like** **is-missing**	01 10 11
Tree / Chain	*(r-id)* + *(r-id)* + **except** + *(c-id)* + **except** + *(c-id)* + ...+ **end statement**	**except** **end statement**	1 0
Fishbone	*(length of block)* + **except** + *(c-id)* + **except** + *(c-id)* + ...+ **end statement**	**except** **end statement**	1 0

TABLE I: Encoding structure used for each method

implemented HYTRA in Matlab; all experiments were carried out on a 2.4 GHz Intel Core i5 Macbook Pro, 16 GB RAM, running OS X 10.9.5. Our code and all our datasets are publicly available at http://www.cs.cmu.edu/~hyunahs/tol. We used 5 groups in the grouping stage for HYTRA-Onion, 100 features ($p = 100$) for HYTRA-Chain and HYTRA-Tree, and threshold $C = 50$ for HYTRA-Fishbone. The synthetic datasets used are:

1) KRONECKER: a 256×256 Kronecker graph [3], i.e. a repeated Kronecker product of a 2 by 2 matrix with itself to produce self-similarity.
2) BLOCKS: a 100×140 matrix containing two 50×70 blocks of ones and another 48×36 block.
3) HYPERBOLIC: a 20×20 matrix containing 3 overlapping communities of sizes 20, 8 and 4, each resembling a scale-free network.

The real datasets used are:

1) ANIMAL: a 34 by 13 animal-property table, originally used in [4] consisting animals and their properties, with 136 animal-property pairs.
2) NELL: a 212 985 by 217 object-category table, containing 1.1 million facts (object-category pairs); the facts are the Never Ending Language Learner's (NELL) annotations of a set of noun phrases using its ontology [5].
3) DRUG-BANK: a 1581 by 16 883 table consisting of 1581 drugs and their properties, with 109 339 substance-property pairs[6].
4) QUESTIONS: a 60 by 218 matrix consisting of the answers by human subjects to 218 simple questions during a brain study, such as 'do airplanes fly?' It has 5252 nonzeros, and has been used in [7] and [8].

Our experiments with HYTRA on various types of synthetic datasets consisting of various data structures (Kronecker matrix, multiple blocks, and hyperbolic degree distributions) demonstrated that each algorithm returns the best result in terms of encoding length and our metric for data structures it is designed for, allowing for flexibility to handle various types of real data (detailed comparisons can be found in our technical report at http://www.andrew.cmu.edu/user/bhooi/hytra.pdf).

Our experiments with real datasets demonstrated that it scales linearly with the input(**scalability**). Figure 2 compares HYTRA to the baseline method 'Dot by Dot', the typical method of encoding sparse data by encoding the row and column indices of each nonzero entry in the matrix, showing that HYTRA encodes data more efficiently, both under total encoding length and *ALOC* (**effectiveness**).

As we showed in Figure 3, a by-product of HYTRA is the automatic reordering and grouping of the data, which can be interpreted as teaching order found by HYTRA (**intuitiveness**). This reordering and grouping is illustrated in Figure 3 (bottom), in which HYTRA reorders the matrix so as to group related entities in an intuitive way.

V. CONCLUSION

In this paper, we considered the problem of teaching a collection of facts while minimizing student effort. Our contributions are as follows:

- **Problem Formulation**: we define the problem of transmitting a matrix of objects and properties adhering to principles from the theory of (human) learning.
- **Optimization Goal**: We define an appropriate optimization goal, namely minimizing *ALOC*, and explain how it corresponds to minimizing student effort and maximizing student utility.
- **Algorithm**: We propose HYTRA, a multi-pronged method that encodes the data while reordering and grouping the data. We evaluate HYTRA on synthetic and real datasets, showing that it encodes data more efficiently than a standard approach for encoding sparse data, measured using both *ALOC* and total encoding length. On real datasets, we find that the orderings and groupings it produces are meaningful.

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under Grant No. IIS-1247489. Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053.

This work is also partially supported by an IBM Faculty Award and a Google Focused Research Award. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, or other funding parties. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

REFERENCES

[1] J. Feldman, "Minimization of boolean complexity in human concept learning," *Nature*, vol. 407, no. 6804, pp. 630–633, 2000.

[2] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

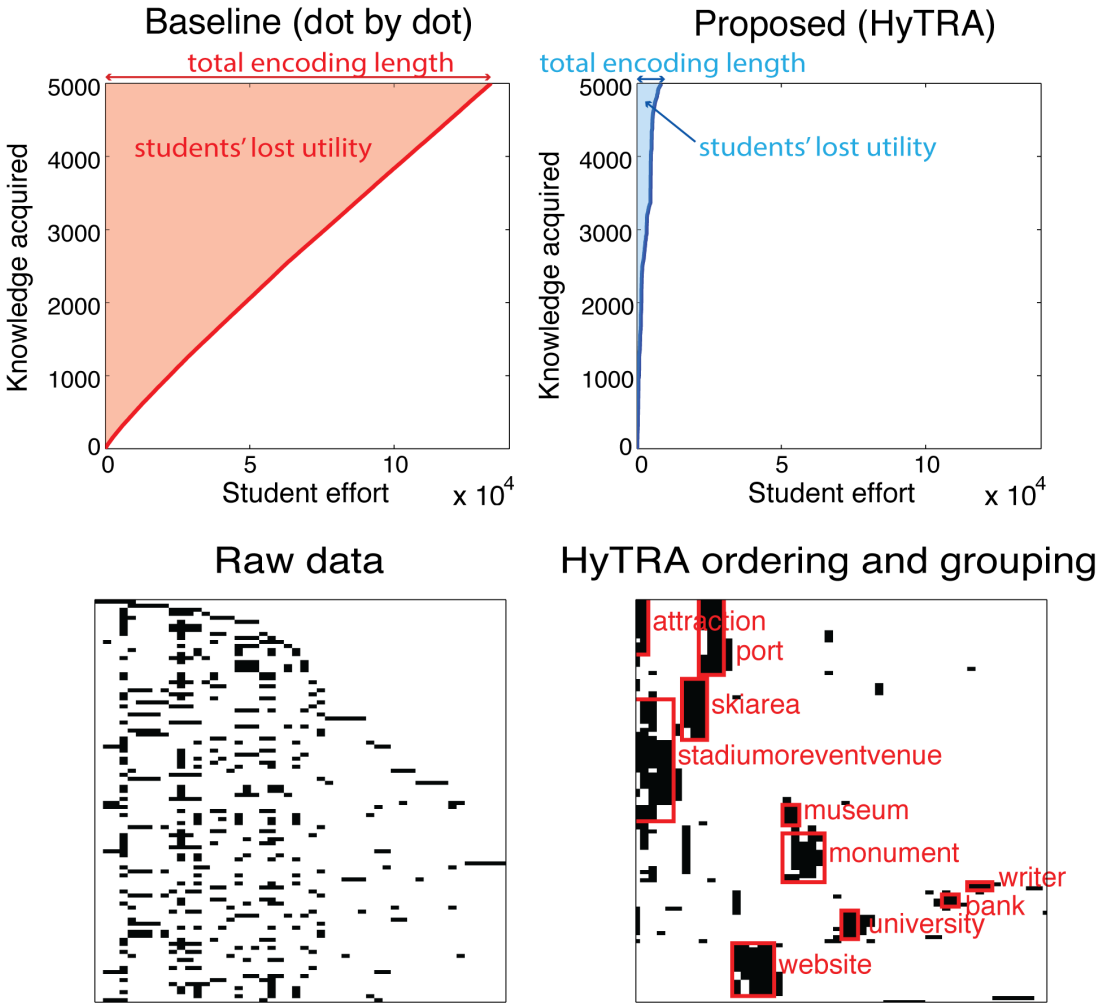

Fig. 3: Results of HyTRA on facts dataset obtained from the Never-Ending Language Learner: rows represent particular places (e.g. Iron Range National Park) while columns represent categories (e.g. museum). **Top:** HyTRA encodes facts with much lower total encoding length than the dot by dot baseline encoding, which encodes the nonzero entries of the matrix one by one. The shaded areas are a measure of students' lost utility, as we explain in our Proposed Metric section, and HyTRA also does much better by this measure. **Bottom:** Additionally, as a side effect, HyTRA finds a reordering and groupings of the facts which are intuitive and interpretable in practice. Groupings (red rectangles) as well as labels were generated automatically by HyTRA.

[3] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, "Kronecker graphs: An approach to modeling networks," *JMLR*, vol. 11, pp. 985–1042, 2010.

[4] R. Bro, E. E. Papalexakis, E. Acar, and N. D. Sidiropoulos, "Coclustering - a useful tool for chemometrics," *Journal of Chemometrics*, vol. 26, no. 6, pp. 256–263, 2012.

[5] "Read the web," http://rtw.ml.cmu.edu/rtw/.

[6] C. Knox, V. Law, T. Jewison, P. Liu, S. Ly, A. Frolkis, A. Pon, K. Banco, C. Mak, V. Neveu *et al.*, "Drugbank 3.0: a comprehensive resource for omics research on drugs," *Nucleic acids research*, vol. 39, no. suppl 1, pp. D1035–D1041, 2011.

[7] B. Murphy, P. Talukdar, and T. Mitchell, "Selecting corpus-semantic models for neurolinguistic decoding," in *ACL *SEM*. Association for Computational Linguistics, 2012, pp. 114–123.

[8] E. E. Papalexakis, T. M. Mitchell, N. D. Sidiropoulos, C. Faloutsos, P. P. Talukdar, and B. Murphy, "Turbo-smt: Accelerating coupled sparse matrix-tensor factorizations by 200x," in *SDM*, 2014.