Mining Large Multi-Aspect Data: Algorithms & Applications

Evangelos E. Papalexakis
epapalex@cs.cmu.edu
School of Computer Science
Carnegie Mellon University
Advisor: Prof. Christos Faloutsos

ABSTRACT

Given a Knowledge Base that records millions of relations of the form Barack Obama is the president of USA, how can we automatically learn new synonyms and enhance the Knowledge Base? Imagine now measuring the brain activity of a person while reading words that appear in this Knowledge Base; how can we relate information processing in the brain, and information found on the World Wide Web? Can we use both pieces of data in order to enhance knowledge extraction in both scenarios? On a third, seemingly unrelated, application, consider having different views of a social network, e.g. observing who is calling whom, who sends e-mails to whom, and who texts whom; can we use this rich information towards community and anomaly detection? What if we also have demographic information about the people of the network? Can we further enhance our analysis? The key underlying theme behind all the above applications is the multi-aspect nature of the data, with the ultimate question being: how can we take advantage of all different aspects? And if so, can we analyze sets of multi-aspect data jointly? Finally, can we automatically, and in a mostly unsupervised setting, filter out aspects of the data which are redundant or not beneficial for the task at hand? In this thesis, we develop fast, scalable, and interpretable algorithms (with specific emphasis on Tensor Analysis), and we apply them to a wide variety of multi-aspect data problems.

1. INTRODUCTION

In an ever increasing number of real world applications, data produced come in different views or aspects, often describing a common underlying phenomenon. Such a phenomenon can be, for instance, the way that knowledge is manifested on the Web. Consider a Knowledge Base (KB) such as NELL [1] that reads the web every day and learns new facts about the world. This KB is expressed in millions of (subject, verb, object) triplets, like Barack Obama is the president of USA; essentially *subject*, *verb*, and *object* are the three aspects of the data, and our aim is to use all three jointly in order to learn new synonyms and ultimately enhance the KB. Suppose now, that for words that exist in that KB, we measure a person's brain activity while reading each word. How can we come up with effective, structured, and principled ways of relating the information as it is manifested on the KB and the Web on the one hand, and the signals of the human brain in the presence of that information? Furthermore, how can we improve knowledge extraction and understanding of both processes, by using both sides of the data?

Another domain which is inherently multi-aspect, is the one of social networks, especially with the proliferation of online social networks such as Facebook. Different means of communication yield different views of a social network: for instance, the social network of people who call each other and the social network of people who e-mail or message each other are different aspects of the same underlying social interaction on that set of people. How can we use these different aspects in order to better understand the social interactions of the underlying network? Suppose now that we also have rich side information about the people of the network. How can we incorporate this side information in our analysis, in order to further improve our results?

The unifying theme behind the above, seemingly unrelated applications, is the multi-aspect nature of the data. In this thesis, we work towards in two different thrusts:

Algorithms: we develop multi-aspect analysis models and scalable algorithms, with specific emphasis to Tensor Analysis, that are able to efficiently extract knowledge from multi-aspect data. Our motivating questions are how can we take advantage of all different aspects? And if so, can we analyze sets of multi-aspect data jointly? Finally, can we automatically, and in a mostly unsupervised setting, filter out aspects of the data which are redundant or not beneficial for the task at hand?

Applications: we apply our algorithms to a variety of multi-aspect data problems, with specific emphasis on linking knowledge extraction from the Web and the brain, as well as analyzing multi-aspect social networks.

2. PRELIMINARIES

Our methods have a specific emphasis on Tensor analysis. Thus, here, we provide a very brief, high level overview of how Tensors can be used as an exploratory analysis tool, using as a motivating example that of a Knowledge Base. Tensor analysis is by no means a new area, however, our on-going and proposed work is novel in the context of the applications that we are interested in, as well as the new models and algorithms that we develop.

Matrices record dyadic properties, like "people recommending products". Tensors are the n-mode generalizations, capturing 3- and higher-way relationships. Effectively, Tensors can be seen as a multi-dimensional extension of matrices. For example "subject-verb-object" triplets naturally lead to a 3-mode tensor. In this overview we focus on three mode tensors, however, everything we mention extends directly to higher modes

Proceedings of the VLDB 2015 PhD Workshop

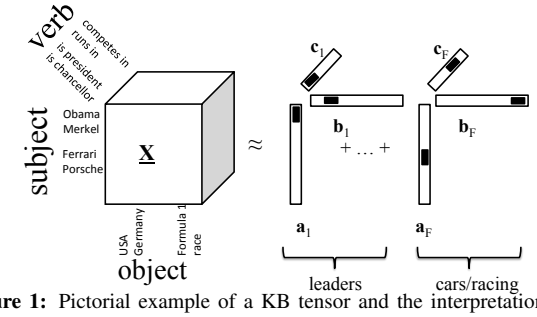


Figure 1: Pictorial example of a KB tensor and the interpretation of its PARAFAC decomposition into sparse factors. The shaded part of the vector corresponds to non-zero values. For the first component, the non-zeros correspond to subjects like "Obama" and "Merkel", and the respective objects and verbs of that component collectively describe a latent group of "leaders". Accordingly, the F-th component is a latent group about cars and racing.

Tensor decomposition as soft clustering:. For instance, given a "subject-verb-object" tensor, one may decompose it into a sum of a (usually) small number of triplets of vectors; intuitively, each one of these triplets corresponds to a different concept, e.g., "politicians", "countries", and "tools". Each vector of this triplet may be viewed as a soft clustering indicator: suppose that $\mathbf{a}, \mathbf{b}, \mathbf{c}$ are the vectors of the "politicians" triplet that correspond to the "subject", "verb" and "object" dimensions (or modes) respectively. Then, $\mathbf{a}$ will indicate the *membership* of all the subjects to the "politicians" cluster, and $\mathbf{b}$ and $\mathbf{c}$ will do so for all the verbs and objects.

For example, see Figure 1. The triplet of vectors $\mathbf{a}_1$, $\mathbf{b}_1$, $\mathbf{c}_1$ will correspond to the first concept (e.g., "leaders-organizations"); subjects (rows) with high score on $\mathbf{a}_1$ will be the leaders, like "obama", "merkel", "eric-schmidt", objects (columns) with high score on $\mathbf{b}_1$ will be organizations, like "usa", "germany", "google", and verbs (fibers) with high score on $\mathbf{c}_1$ will be verbs, like "lead", "is-president-of", and "is-CEO-of".

The PARAFAC decomposition [5] of $\underline{\mathbf{X}}$ into F components is $\underline{\mathbf{X}} \approx \sum_{f=1}^{F} \mathbf{a}_f \circ \mathbf{b}_f \circ \mathbf{c}_f$, where $[\mathbf{a} \circ \mathbf{b} \circ \mathbf{c}](i,j,k) = \mathbf{a}(i)\mathbf{b}(j)\mathbf{c}(k)$.

Coupled Tensors. Sometimes, two tensors, or a matrix and a tensor, may have one mode in common; for example, we may have a 'subject-verb-object' tensor and a 'subject-category' matrix (which encodes the categories where each of the subjects belongs to). In this case, we say that the matrix and the tensor are *coupled* in the 'subjects' mode.

Why Tensors & Coupled Tensors?. There is a number of reasons why we prefer using higher-order structure rather than aggregating/collapsing into a matrix: 1) Tensor decompositions (and in particular the PARAFAC decomposition) are provably unique and identifiable, in contrast to the majority of matrix factorizations. Identifiability implies recovery of the *true* latent factors (e.g. in the case of NELL the cluster assignments of noun-phrases to concepts), without distortions and ambiguities 2) Consider a $10 \times 10 \times 10$ tensor. If we aggregate the third mode into a 10×10 matrix or unfold the tensor into a 10×100 matrix there is *no way* that we can extract more than 10 components uniquely, even though our data might have more structure; in the tensor case this is possible. 3) In the case of coupling, the benefit is twofold: a) additional information from the matrix helps "fill in the blanks" of the tensor (e.g. *cold-start problem* in recommendation systems), b) decomposing the matrix by itself is (albeit widely studied) a less well behaved problem; coupling the matrix with the tensor, *guides* the decomposition into a solution which is more likely to be more well behaved in terms of indeterminancies.

3. COMPLETED & ON-GOING WORK

3.1 Applications

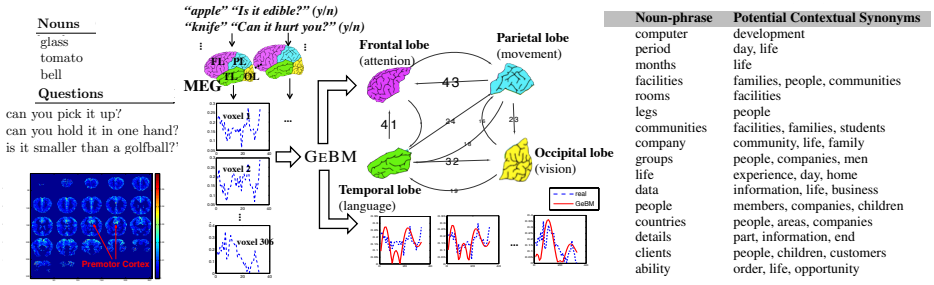
Here, we provide a concise overview of the Applications that we have been tackling using our proposed techniques.

Neurosemantics: Consider the following experimental setting, where human subjects are shown a concrete English noun, and in the meanwhile, we measure their brain activity as they read and try to understand that noun. Our goal is to come up with models that may improve our understanding of how the human brain stores and processes semantic information.

In [12], we *coupled* fMRI measurements of the above experiment with semantic features (in the form of simple questions, such as *Can you pick it up?*) for the same set of nouns; in our analysis, we were able to compute a joint low-rank embedding of the brain measurements and the noun semantic features, discovering semantically similar nouns and coherent brain regions that respond when these nouns are seen. An example of our analysis can be seen in Figure 2(a), where all the nouns are small objects, the corresponding questions reflect holding or picking such objects up, and most importantly, the brain region that was highly active for this set of nouns and questions was the *premotor cortex*, which is associated with holding or picking small items up. In a similar experimental setting, where the human subjects are also asked to answer a simple yes/no question about the noun they are reading, in [11] we define a simple yet effective model that is able to capture the *functional connectivity* of the brain for the particular task; the functional connectivity is a graph between different regions of the brain that interact with each other (and are not necessarily directly physically connected), while the brain processes the semantic information. An example of our derived functional connectivity, which corresponds to Neuroscientific ground truth, is shown in Figure 2(b)

Knowledge Base: A second major application, as also motivated in the previous section is the analysis and expansion of a Knowledge Base, such as the one of the Never Ending Language Learner (NELL) of the *Read the Web* project at CMU. The ability to represent such Knowledge Base data as a three-mode tensor enables the analysis of the data into low-rank embeddings that promote the discovery of synonyms. In the case of a (subject, verb, object) tensor, the low rank embeddings of the corresponding aspects will be $\mathbf{A}, \mathbf{B}, \mathbf{C}$. As we illustrate in the Introduction and Fig. 1, the columns of these low-rank embeddings can serve as soft-clustering indicators, for semantically similar triplets of (subjects, verbs, objects). Furthermore, using those embeddings, we can discover contextually similar nouns, such as the ones shown in Figure 2(c). We have done preliminary work on Knowledge Base mining in [13, 8], however, as we point out in the proposed work, there is still a lot to be done.

Multi-Aspect Social Networks: As mentioned in the Introduction, consider multi-aspect measurements of a social network; different aspects can be *time* or *different views* of the network. In [10] we show that, in general, having different views of a particular social network (e.g. who-texts-whom, who-emails-whom etc) is able to do better community detection than the single view approach, where all types of interactions are aggregated into a single graph/matrix. In [10], we also provide a data mining case study on the REALITYMINING dataset. This dataset was introduced in [4]



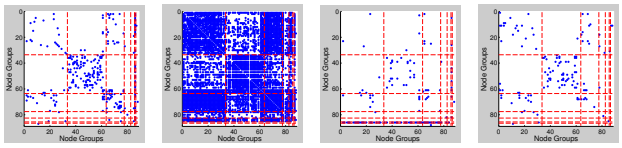
(a) (b)

Noun-phrase	Potential Contextual Synonyms
computer	development
period	day, life
months	life
facilities	families, people, communities
rooms	facilities
legs	people
communities	facilities, families, students
company	community, life, family
groups	people, companies, men
life	experience, day, home
data	information, life, business
people	members, companies, children
countries	people, areas, companies
details	part, information, end
clients	people, children, customers
ability	order, life, opportunity

(c)

Figure 2: Overview of results: (a) Semantically similar nouns and activated brain regions, (b) Computing the *functional connectivity* of the human brain, (c) discovering contextually similar noun-phrases from the *Read the Web* Knowledge Base.

and contains data collected by the MIT Media Lab, including subjects (undergraduate and graduate CS and business students) whose interactions were monitored by a pre-installed piece of software on their mobile devices. The different views offered by the dataset pertain to the means of interaction between a pair of subjects. Namely, `CALL` view refers to subjects calling each other, `DEVICE` view contains Bluetooth device scans, `SMS` view is constructed based on text message exchanges, and `FRIEND` view contains friendship claims. In Fig.3, we show all four views of the dataset as clus-



(a) `CALL` (b) `DEVICE` (c) `SMS` (d) `FRIEND`

Figure 3: Results on the four views of the REALITYMINING multi-graph. Red dashed lines outline the clustering found by GRAPHFUSE.

tered by GRAPHFUSE where $R = 6$. Qualitatively, we see that the algorithm's output concurs with the communities that appear to be strong on the spy-plots of each view. For example, cluster 2 is a community of business school students that are mostly isolated from the rest of the graph. Another example is cluster 6 of size 1, which contains a single subject with many incoming calls and many outgoing SMSs.

In [8], we analyze a time-evolving snapshot of Facebook, where we record users posting to other users' "Wall" ; the temporal aspect is very important in this case, since it helps differentiate types of behavior. For instance, one of the patterns that our Tensor analysis was able to uncover was behavior that looked like a singular event, such as the Wall owner's birthday, where many people posted on the Wall on a single day; ignoring or aggregating the temporal aspect would have made discovery of such events much more difficult, if not impossible.

3.2 Algorithms

With the vast amounts of potential data that can be analyzed using these techniques (and producing beneficial results for the respective applications), major challenges such as *efficiency* and *scalability* arise. We need algorithms that are able to work on data that spill beyond the main memory of a single machine. In [13] we develop the first scalable algorithm for tensor decompositions on Map/Reduce; at the time of publication, [13] was able to decompose problems larger by at least **two orders of magnitude** than the state of the art. Subsequently, in [2], we developed a Distributed Stochastic Gradient Descent method for Map/Reduce that is able to scale to billions of parameters.

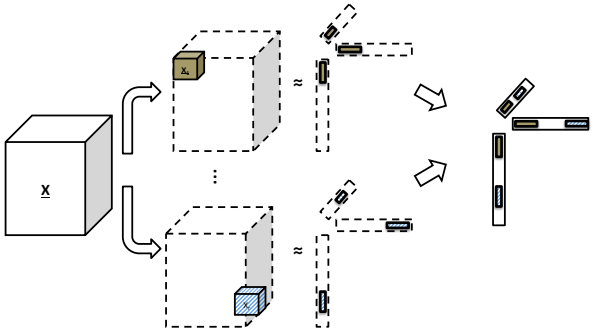


Figure 4: The main idea behind PARCUBE: Using biased sampling, extract small representative sub-sampled tensors, decompose them in parallel, and merge the final results into a set of sparse latent factors.

Not necessarily being restricted to the Map/Reduce framework, in [8] we propose PARCUBE, a novel, approximate, parallelizable algorithm for tensor decomposition which is able to analyze very big tensors on a single, potentially multi-core, workstation. The *main idea* behind PARCUBE is the following

1. Get a *biased sample* of indices of $\underline{\mathbf{X}}$ in all three modes. Biased sampling gives priority to *denser* regions in the data. Every time we do that, we get a smaller sub-tensor, indexed by the set of sampled indices, as shown in Figure 4. Usually the sampled indices per mode are one or more orders of magnitude smaller in size than the original dimension.
2. For each of the smaller sub-tensors, we run the decomposition *in parallel*. In this step we may use any solver for the sub-problem, as long as the solver guarantees a locally optimal solution for the problem.
3. We merge the partial factors coming from the decompositions of the sub-tensors. Notice in Fig. 4 that indices that were not sampled, are shown in white in the final result, indicating that they are exactly equal to 0.

In [8] we describe in detail how we can do that correctly, and obtain a decomposition in the original, un-sampled space, that approximates the full decomposition. The power behind PARCUBE is that, even though the tensor itself might not fit in memory, we can choose the sub-tensors appropriately so that they fit in memory, and we can compensate by extracting many independent sub-tensors. Due to space limitations, we encourage the reader to read

[8] for a detailed experimental evaluation. In a nutshell, we observe that for a small number of repetitions of PARCUBE, there is an understandable gap between the ideal approximation error of PARAFAC and the approximation PARCUBE gives, but as we run more repetitions, we explore the data more effectively, converging to the same approximation error. In [12], we extend the idea of [8], introducing TURBO-SMT, for the case of Coupled Matrix-Tensor Factorization (CMTF), achieving up to **200 times faster** execution with comparable accuracy to the baseline, on a single machine. An important aspect of both PARCUBE and TURBO-SMT is that they can serve as *meta-algorithms* that can boost any already highly optimized state of the art solver for PARAFAC and CMTF; this is because, as illustrated in Fig. 4, each of the smaller sampled pieces of the data can be decomposed by any solver.

4. PROPOSED WORK

Given the spectrum of Applications and Algorithms that we consider, the space of possible extensions that one could explore is promisingly rich and interesting. In the next few lines, we describe some key future directions of our work:

Unsupervised Quality Assessment:. For the most part, our analysis is *unsupervised*, in the sense that we don't have labelled data or ground truth for the knowledge that we wish to extract; in other words, our analysis is largely *exploratory*. However, we would like to have ways of assessing the quality of our results in absence of ground truth. A particular example where this is of paramount importance is the following: given different views of a social network, how can we automatically detect whether a particular view is offering *useful* information or is merely noise? There exist heuristics in the literature [3] which are able to do well in determining the number of hidden components in a tensor (even though this has been shown to be a very hard problem). However, these heuristics have been specifically designed for fully dense, relatively small datasets, where the fitting is done under the Frobenius norm. As a first step, we propose to extend these intuitive heuristics to scale and be able to work for very large and sparse datasets (such as social networks). We have recently published preliminary algorithmic work on this [7], where we are able to work on **three orders of magnitude larger data** than the state of the art. Secondly, we may consider applying the Minimum Description Language (MDL) principle in order to characterize the quality of a decomposition, as well as approximate the true number of hidden components.

Robust Knowledge Base Completion & Synonym Discovery:. Triplets of a Knowledge Base reflect what the Knowledge Base already knows about the world. Triplets that are missing from the Knowledge Base could be missing for more than one reasons: they could either be unobserved but *plausible* (e.g. *horses eat hay*) or unobserved but *implausible* (e.g. *horses eat cars*). If we treat all unobserved values as missing (and thus, suitable for completion), our results will likely suffer from this ambiguity. We plan to investigate robust ways of overcoming this real world problem.

Location Based Social Networks. Location Based Social Networks (LBSNs) are services such as Foursquare, that are primarily focused on facilitating location sharing among their users. Such a location sharing involves a user "checking-in" at a specific venue. Venues can be businesses, public places, even a user's home. Check in activity is sometimes associated with rewards from specific businesses, like restaurants, thus there is incentive by users to increase their number of check-ins at a place that offers a specific discount in fraudulent ways. We have applied our algorithms in detecting anomalies in various scenarios [8, 6], and we propose to investigate how our algorithms can be applied in order to detect fraudulent check-ins in LBSNs, as they evolve over time.

In addition to fraud detection, location information which is an integral part of LBSNs provides very rich information that can be used for user modelling. More specifically, given a user's check-in activity, we may be able to provide better recommendations for places to visit, as well as better friendship recommendations, based on similar preferences in terms of, say, restaurants, coffee shops and bars. We propose to model a LBSN as a tensor of (users, locations/venues, time) and a matrix of user friendships that can serve as additional information. Using our proposed algorithms, we can then jointly analyze these two pieces of data, into a comprehensive user model that takes into account time, location and friendship relations. Our very preliminary results were presented as a poster in WWW 2014 [9].

5. ACKNOWLEDGMENTS

Research was supported by the National Science Foundation Grant No. IIS-1247489. 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 funding parties.

6. REFERENCES

[1] Read the web. `http://rtw.ml.cmu.edu/rtw/`.

[2] A. Beutel, A. Kumar, E. Papalexakis, P. P. Talukdar, C. Faloutsos, and E. P. Xing. Flexifact: Scalable flexible factorization of coupled tensors on hadoop. 2014.

[3] R. Bro and H. A. Kiers. A new efficient method for determining the number of components in parafac models. *Journal of chemometrics*, 17(5):274–286, 2003.

[4] N. Eagle, A. Pentland, and D. Lazer. Inferring friendship network structure by using mobile phone data. *PNAS*, 106(36):15274–15278, 2009.

[5] R. Harshman. Foundations of the parafac procedure: Models and conditions for an" explanatory" multimodal factor analysis. 1970.

[6] H.-H. Mao, C.-J. Wu, E. E. Papalexakis, C. Faloutsos, K.-C. Lee, and T.-C. Kao. Malspot: Multi2 malicious network behavior patterns analysis. In *Advances in Knowledge Discovery and Data Mining*, pages 1–14. Springer, 2014.

[7] E. Papalexakis and C. Faloutsos. Fast efficient and scalable core consistency diagnostic for the parafac decomposition for big sparse tensors. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015.

[8] E. Papalexakis, C. Faloutsos, and N. Sidiropoulos. Parcube: Sparse parallelizable tensor decompositions. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 2012.

[9] E. Papalexakis, K. Pelechrinis, and C. Faloutsos. Spotting misbehaviors in location-based social networks using tensors. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, pages 551–552. International World Wide Web Conferences Steering Committee, 2014.

[10] E. E. Papalexakis, L. Akoglu, and D. Ienco. Do more views of a graph help? community detection and clustering in multi-graphs. In *Information Fusion (FUSION), 2013 16th International Conference on*, pages 899–905. IEEE, 2013.

[11] E. E. Papalexakis, A. Fyshe, N. D. Sidiropoulos, P. P. Talukdar, T. M. Mitchell, and C. Faloutsos. Good-enough brain model: challenges, algorithms and discoveries in multi-subject experiments. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 95–104. ACM, 2014.

[12] 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 *SIAM SDM*, 2014.

[13] K. U, P. E.E., H. A., and F. C. Gigatensor: Scaling tensor analysis up by 100 times - algorithms and discoveries. In *Proceedings of the 18th international ACM SIGKDD conference on Knowledge Discovery and Data Mining*. ACM, 2012.