Mining Large Multi-Aspect Data: 
Algorithms and Applications

Evangelos E. Papalexakis

February 13, 2015

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Christos Faloutsos, Chair
Tom Mitchell
Jeff Schneider
Nicholas D. Sidiropoulos, University of Minnesota

Submitted in partial fulfillment of the requirements 
for the degree of Doctor of Philosophy.

Copyright © 2015 Evangelos E. Papalexakis
Keywords: multi-aspect data mining, tensors, tensor decompositions, knowledge base, brain data mining, multi-view social networks, time-evolving social networks, scalability, interpretability
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 (KB); 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; for instance, in the social network case, each view of the network is a different aspect of the data. Thus, the ultimate question becomes: 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 work towards answering the above questions, 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.
- **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.

In our completed work we have already made progress in all three major applications; with respect to Algorithms, we have produced methods that are up to two orders of magnitude faster than the state of the art, while maintaining accuracy comparable or equal to the state of the art and results up to 90% sparser.
## Contents

1 Introduction ................................................................. 1
   1.1 Introduction ......................................................... 1
   1.2 Completed Work .................................................... 2
      1.2.1 Applications .................................................. 2
      1.2.2 Algorithms .................................................... 3
   1.3 On-going & Proposed Work ........................................... 3
      1.3.1 Applications .................................................. 3
      1.3.2 Algorithms .................................................... 4
   1.4 Reproducibility and Impact ......................................... 4
   1.5 Overview .............................................................. 4

2 Preliminaries ............................................................... 5

3 Completed Work: Applications ............................................. 7
   3.1 App1: Neurosemantics: ............................................. 7
      3.1.1 Exploratory Analysis of Brain Activity Coupled with Semantic Information ................................................. 8
      3.1.2 Predicting Brain Activity from Questions ............................ 8
      3.1.3 Modelling the Functional Connectivity of the Brain .................. 9
      3.1.4 Multi-subject Functional Connectivity .............................. 11
   3.2 App2: Knowledge Base ................................................ 11
   3.3 App3: Multi-Aspect Social Networks ................................ 13
      3.3.1 Multi-View Social Networks ..................................... 14
      3.3.2 Time-Evolving Social Networks .................................. 14

4 Completed Work: Algorithms ............................................... 17
   4.1 Alg1: Scalability & Speed ........................................... 17
   4.2 Alg2: Interpretability ................................................ 19

5 On-going & Proposed Work ................................................ 21
   5.1 Applications ........................................................... 21
      5.1.1 App1: Modelling the Functional Connectivity of the Brain ................................................................. 21
      5.1.2 App2: Robust Knowledge Base Completion & Synonym Discovery ......................................................... 22
      5.1.3 App3: Multi-Aspect Social Networks .............................. 22
      5.1.3.1 Location Based Social Networks ................................ 22
      5.1.3.2 Analyzing Bilingual Immigrant Communities .................. 23
   5.2 Algorithms ............................................................. 23
5.2.1 Alg2: Interpretability .............................................. 23
5.2.1.1 Unsupervised Quality Assessment ......................... 24
5.2.1.2 Mode Redundancy & Structural Abnormalities ............ 24

6 Related Work .......................................................... 26
6.1 Tensor Decompositions and Applications ....................... 26
6.2 Coupled, Multi-block, Multi-set Models ......................... 27
6.3 Fast & Scalable Tensor Decompositions ......................... 27

7 Timeline ................................................................. 29

8 Conclusion ............................................................... 30

Bibliography ............................................................. 31
Chapter 1

Introduction

1.1 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 in 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.

1.2 Completed Work

For the sake of motivation, we will first outline the completed work in terms of Applications and secondly the work in terms of Algorithms. We will follow the same order when outlining the proposed work.

1.2.1 Applications

Our applications range from linking knowledge extraction from the Brain and the Web, to analyzing multi-aspect/view social networks. In [47][PDF] we identify regions of the brain that get activated for semantically similar concepts. Furthermore, in [46][PDF], we introduce a simple and effective model for identifying the functional connectivity of the brain 1, inspired by Control Theory. Another major application is that of Knowledge Base (KB) mining and completion; in [59][PDF], [41][PDF] we have done preliminary work, extracting contextual synonyms from a very large KB[1] by using tensor decompositions to compute low rank word embeddings. On the topic of multi-aspect social network analysis, we have worked towards community and anomaly detection in time-evolving social networks [41][PDF], [9][PDF] (on anomaly detection we also have work on intrusion detection systems [38][PDF]) and Location Based Social Networks [44][PDF]. Finally, in [45][PDF] 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 view/matrix.

1Functional connectivity, in a nutshell, is defined as the communication patterns between different parts of the brain, when a person is executing a specific task, like reading.
1.2.2 Algorithms

We develop multi-aspect analysis models and scalable algorithms, with specific emphasis to Tensor Analysis. With the vast amounts of potential data that can be analyzed using these techniques, 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 [59][PDF] [16][PDF] we introduce highly scalable Map/Reduce based tensor decomposition algorithms. In [41][PDF] [47][PDF], we introduce highly parallelizable approximate algorithms that can analyze very large datasets, on a single machine, and up to 200 times faster than the state of the art. We formally show in [49][PDF] that approximate approaches of similar flavor can, in fact, guarantee identifiability under appropriate conditions. Besides scalability, we work towards interpretability; The algorithms in [41][PDF] [47][PDF] produce sparse factors which are easily interpretable.

Figure 1.2: (a): Big picture behind PARCUBE [41] & TURBO-SMT [47]: reduce dimensionality through sampling, parallelize, and merge partial results (b): TURBO-SMT is up to 200x faster, for comparable accuracy. Relative execution time vs relative cost (lower is better), for various settings. TURBO-SMT boosting either ALS or CMTF-OPT [4]) (i.e. the baselines) vs. plain execution of the baselines (on a single core).

1.3 On-going & Proposed Work

Here, we illustrate the main research questions behind each proposed task. Details on each task can be found in the respective subsections.

1.3.1 Applications

Modelling the Functional Connectivity of the Brain (§5.1.1) Can we apply models such as the Recurrent Neural Network in modelling the functional connectivity of the brain?

Robust Knowledge Base Completion & Synonym Discovery (§5.1.2) In the context of modelling a KB as a tensor: How can we decide whether we should complete a missing value in the KB or not? Can we incorporate type information in KB completion?

Location Based Social Networks (§5.1.3.1) Can we detect normal and abnormal behavior from social data with location information? Can we use location information to improve friend recommendation, and friendship information to improve venue recommendation?
Analyzing Bilingual Immigrant Communities (§5.1.3.2) Can we automatically identify bilingual users in immigrant communities? What are the topics where bilingual users code-switch the most? Are users strictly bilingual or monolingual, or does it depend on factors such as the discussion topic, the social context, or the time?

1.3.2 Algorithms

Unsupervised Quality Assessment (§5.2.1.1) Given a large and potentially very sparse tensor, and its PARAFAC decomposition, is the decomposition meaningful? Given a large and potentially very sparse tensor, what is a good number of components for its PARAFAC decomposition?

Mode Redundancy & Structural Abnormalities (§5.2.1.2) Given structurally problematic data, can we derive robust methods that are able to extract useful information despite these imperfections?

1.4 Reproducibility and Impact

Algorithms produced throughout this work are being made publicly available through http://www.cs.cmu.edu/~epapalex/code.html, promoting reproducibility of the results and re-usability of our algorithms. At the time of writing of the present proposal:

- PARCUBE & TURBO-SMT had been downloaded by around 30 institutions in 13 countries,
- GRAPHFUSE had been downloaded by around 30 institutions in 13 countries,
- GeBM has been downloaded by around 20 institutions in 8 countries.

1.5 Overview

Table 1.1 shows a breakdown of our work for both tasks (Applications and Algorithms), in terms of completed and proposed work. For the reader’s convenience, each hyperlink to a PDF file points to an online version of the corresponding paper.

<table>
<thead>
<tr>
<th>Application Tasks (App)</th>
<th>Completed Work</th>
<th>On-going &amp; Proposed Work</th>
</tr>
</thead>
<tbody>
<tr>
<td>App1: Neurosemantics (§3.1) [47][PDF] [46][PDF]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>App2: Knowledge Base (§3.2) [59][PDF]</td>
<td>App1: Modelling the Functional Connectivity of the Brain (§5.1.1)</td>
<td></td>
</tr>
<tr>
<td>App3: Multi-Aspect Social Networks (§3.3) [41][PDF] [45][PDF] [9][PDF]</td>
<td>App2: Robust Knowledge Base Completion &amp; Synonym Discovery (§5.1.2)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithm Tasks (Alg)</th>
<th>Completed Work</th>
<th>On-going &amp; Proposed Work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alg1: Scalability &amp; Speed (§4.1) [59][PDF] [16][PDF] [41][PDF] [47][PDF] [49][PDF]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alg2: Interpretability (§4.2) [41][PDF] [47][PDF] [6][PDF]</td>
<td>Alg2: Unsupervised Quality Assessment (§5.2.1.1) Alg2: Mode Redundancy &amp; Structural Abnormalities (§5.2.1.2)</td>
<td></td>
</tr>
</tbody>
</table>

Table 1.1: Breakdown of completed and proposed work
Chapter 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 problem, 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. For example “subject-verb-object” triplets naturally lead to a 3-mode tensor.

\[
\begin{align*}
X & \approx \text{objects} \\
& \quad \text{subjects} \\
& \quad \text{verbs} \\
& \quad \text{Concept1} \\
& \quad \text{Concept2} \\
& \quad \text{Concept R} \\
\end{align*}
\]

Figure 2.1: PARAFAC decomposition of a three-way tensor of the NELL dataset as sum of \( R \) outer products (rank-one tensors), reminiscent of the rank-\( R \) singular value decomposition of a matrix. Each component corresponds to a latent concept, e.g. ”leaders”, ”cars”, ”tools” and so on.

**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 \( a, b, c \) are the vectors of the “politicians” triplet that correspond to the “subject”, “verb” and “object” dimensions (or modes) respectively. Then, \( a \) will indicate the membership of all the subjects to the “leaders” cluster, and \( b \) and \( c \) will do so for all the verbs and objects.

For example, see Figure 2.1. The triplet of vectors \( a_1, b_1, c_1 \) will correspond to the first concept (e.g., “leaders-organizations”); subjects/rows with high score on \( a_1 \) will be the leaders, like “obama”, “merkel”, “eric-schmidt”, objects/columns with high score on \( b_1 \) will be organizations, like “usa”, “germany”,...
“google”, and verbs/fibers with high score on $c_1$ will be verbs, like “lead”, “is-president-of”, and “is-CEO-of”.

The PARAFAC decomposition [30] of $X$ into $F$ components is $X \approx \sum_{f=1}^{F} a_f \circ b_f \circ c_f$,

where $[a \circ b \circ c](i,j,k) = a(i)b(j)c(k)$.

**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.

![Coupled Tensor Example](image)

(a) coupled tensor-matrix  
(b) coupled tensors

Figure 2.2: Coupled tensor example: Tensors often share one or more modes (with thick, wavy line):

In this work we focus on three mode tensors, however, everything we mention extends directly to higher modes. In the general case, a three mode tensor $X$ may be coupled with at most three matrices $Y_i$, $i = 1 \cdots 3$, in the manner illustrated in Figure 2.2 for one mode. The optimization function that encodes this decomposition is:

$$\min_{A,B,C,D,E,G} \|X - \sum_k a_k \circ b_k \circ c_k\|_F^2 + \|Y_1 - AD^T\|_F^2 + \|Y_2 - BE^T\|_F^2 + \|Y_3 - CG^T\|_F^2$$

where $a_k$ is the $k$-th column of $A$. The idea behind the coupled matrix-tensor decomposition is that we seek to jointly analyze $X$ and $Y_i$, decomposing them to latent factors who are coupled in the shared dimension. For instance, the first mode of $X$ shares the same low rank column subspace as $Y_1$; this is expressed through the latent factor matrix $A$ which jointly provides a basis for that subspace.

**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 $10 \times 10$ matrix or unfold the tensor into a $10 \times 100$ 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.
Chapter 3

Completed Work: Applications

Here, we provide a concise overview of the Applications that we have been tackling using our proposed techniques. In the beginning of every section, we summarize the main goal of the respective task. Whenever higher level of detail in the summary is appropriate, we summarize the problem definition on a subsection level.

3.1 App1: Neurosemantics:

| Main Goal:  | Given brain measurements from multiple subjects, exposed to different semantic stimuli and tasks, extract information that can help our understanding of how the brain works. |

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.

Figure 3.1: Coupling an fMRI tensor of (nouns, voxels, participants) with a matrix that holds semantic features for the nouns of the tensor.
3.1.1 Exploratory Analysis of Brain Activity Coupled with Semantic Information

**Problem Definition:**

**Given:** A (nouns, voxels, persons) tensor $X$ and a (nouns, questions) matrix $Y$  
**Find:** A Coupled Matrix-Tensor Factorization of $X$ and $Y$.  
**To:** Extract latent groups that combine semantically similar nouns, voxels/regions of the brain that get activated for these nouns, and questions that are related to these nouns.

In [47], 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 as shown in Fig. 3.1. Using our highly efficient algorithm TURBO-SMT, we were able to compute a sparse, 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 3.2; most notably, in Group 3, 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.

![Figure 3.2: TURBO-SMT finds meaningful groups of words, questions, and brain regions that are (both negatively and positively) correlated. For instance, Group 3 refers to small items that can be held in one hand, such as a tomato or a glass, and the activation pattern is very different from the one of Group 1, which mostly refers to insects, such as bee or beetle. Additionally, Group 3 shows high activation in the *premotor cortex* which is associated with the concepts of that group.](image)

3.1.2 Predicting Brain Activity from Questions

**Problem Definition:**

**Given:** The Coupled Matrix Tensor Factorization of $X$ and $Y$, a new noun and its semantic features/questions.  
**Find:** A mapping from questions to voxels and vice-versa  
**To:** Predict brain activity that corresponds to that new noun.

In addition to soft-clustering, this low dimensional embedding of the data into a common semantic space,
enables the prediction of, say, the brain activity of a subject, for a given word, given the corresponding vector of question answers for that word. In particular, we use this linear transformation in order to map the question answer vector to the latent semantic space and then expanding it to the brain voxel space, we obtain a fairly good prediction of the brain activity.

To evaluate the accuracy of these predictions of brain activity, we follow a leave-two-out scheme, where we remove two words entirely from the brain tensor and the question matrix; we carry out the joint decomposition, in some very low dimension, for the remaining set of words and we obtain the usual set of matrices $A, B, C, D$. Due to the randomized nature of TURBO-SMT, we did 100 repetitions of the procedure described below.

Let $q_i$ be the question vector for some word $i$, and $v_i$ be the brain activity of one human subject, pertaining to the same word. By left-multiplying $q_i$ with $D^T$, we map $q_i$ to the latent space of the decomposition; then, by left-multiplying the result with $B$, we map the result to the brain voxel space. Thus, our estimated (predicted) brain activity is obtained as $\hat{v}_i = BD^Tq_i$.

Given the predicted brain activities $\hat{v}_1$ and $\hat{v}_2$ for the two left out words, and the two actual brain images $v_1$ and $v_2$ which were withheld from the training data, the leave-two-out scheme measures prediction accuracy by the ability to choose which of the observed brain images corresponds to which of the two words. After mean-centering the vectors, this classification decision is made according to the following rule:

$$\|v_1 - \hat{v}_1\|_2 + \|v_2 - \hat{v}_2\|_2 < \|v_1 - \hat{v}_2\|_2 + \|v_2 - \hat{v}_1\|_2$$

Although our approach is not designed to make predictions, preliminary results are very encouraging: Using only $F=2$ components, for the noun pair closet/watch we obtained mean accuracy of about 0.82 for 5 out of the 9 human subjects. Similarly, for the pair knife/beetle, we achieved accuracy of about 0.8 for a somewhat different group of 5 subjects. For the rest of the human subjects, the accuracy is considerably lower, however, it may be the case that brain activity predictability varies between subjects, a fact that requires further investigation.

### 3.1.3 Modelling the Functional Connectivity of the Brain

#### Problem Definition:

- **Given:** Brain activity measurements for a set of $m$ sensors, over time (vector $y(t)$) for a given stimulus and a task (vector $s(t)$).
- **Find:** The functional connectivity of the brain, i.e. the communication/correlation patterns between different regions of the brain, for the specific stimulus/task.
- **To:** Be able to simulate real brain activity, and ideally correspond to known neuroscientific facts.

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 [46] we define GeBM, 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.

For the particular experimental setting, prior work [54] has only considered transformations from the space of noun features to the voxel space and vice versa, as well as word-concept specific prediction based on estimating the covariance between the voxels [29].

In [46] we model the brain as a linear dynamical system with hidden states. Putting everything together, we end up with the following set of equations, which constitute our proposed model GeBM:
Figure 3.3: Big picture: GEBM estimates the hidden functional connectivity (top right, weighted arrows indicating number of inferred connections), when given multiple human subjects (left) that respond to yes/no questions (e.g., edible?) for typed words (e.g., apple). Bottom right: GEBM also produces brain activity (in solid-red), that matches reality (in dashed-blue).

\[
x(t + 1) = A_{[n \times n]} \times x(t) + B_{[n \times s]} \times s(t) \\
y(t) = C_{[m \times n]} \times x(t)
\]  

(3.1) (3.2)

where \(x(t)\) is the hidden brain activity, \(y(t)\) is the observed brain activity measured via Magnetoencephalography (MEG), and \(s(t)\) is the stimulus signal (which encodes the word shown and the question asked). The rest of the matrices and the key concepts behind GEBM are:

- **(Latent) Connectivity Matrix**: We assume that there are \(n\) regions, each containing 1 or more neurons, and they are connected with an \(n \times n\) adjacency matrix \(A_{[n \times n]}\). We only observe \(m\) voxels, each containing multiple regions, and we record the activity (e.g., magnetic activity) in each of them; this is the total activity in the constituent regions
- **Measurement Matrix**: Matrix \(C_{[m \times n]}\) is an \(m \times n\) matrix, with \(c_{i,j} = 1\) if voxel \(i\) contains region \(j\)
- **Perception Matrix**: Matrix \(B_{[n \times s]}\) shows the influence of each sensor to each neuron-region. The input is denoted as \(s\), with \(s\) input signals
- **Sparsity**: We require that our model’s matrices are sparse; only few sensors measure a specific brain region. Additionally, the interactions between regions should not form a complete graph, and finally, the perception matrix should map only few activated sensors to neuron regions at every given time.

Given matrices \(A, C\) in [46] we show that we can derive the connectivity between the \(m\) sensors, as long as the number of latent “neurons” \(n\) is smaller than the number of sensors \(m\). By doing so, we can map the connectivity patterns to physical regions of the brain. An example of our derived functional connectivity, which corresponds to Neuroscientific ground truth, is shown in Figure 3.3.
3.1.4 Multi-subject Functional Connectivity

<table>
<thead>
<tr>
<th>Problem Definition:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Given:</strong></td>
</tr>
<tr>
<td><strong>Find:</strong></td>
</tr>
<tr>
<td><strong>To:</strong></td>
</tr>
</tbody>
</table>

In our experiments, we have 9 participants, all of whom have undergone the same procedure, being presented with the same stimuli, and asked to carry out the same tasks. Availability of such a rich, multi-subject dataset inevitably begs the following question: are there any differences across people’s functional connectivity? Or is everyone, more or less, wired equally, at least with respect to the stimuli and tasks at hand?

By using GEBM, we are able (to the extent that our model is able to explain) to answer the above question. We trained GEBM for each of the 9 human subjects, using the entire data from all stimuli and tasks, and obtained matrices $A$, $B$, $C$ for each person. For the purposes of answering the above question, it suffices to look at matrix $A$ (which is the hidden functional connectivity), since it dictates the temporal dynamics of the brain activity.

In this multi-subject study we have two very important findings:

- **Regularities:** For 8 out of 9 human subjects, we identified almost identical GEBM instances, both with respect to RMSE and to spectrum. In other words, for 8 out of 9 subjects in our study, the inferred functional connectivity behaves almost identically. This fact most likely implies that for the particular set of stimuli and assorted tasks, the human brain behaves similarly across people.

- **Anomaly:** One of our human subjects (#3) deviates from the aforementioned regular behavior.

In Fig. 3.4(a) & (b) we show the real and imaginary parts of the eigenvalues of $A$. The reason we care about the eigenvalues is because, from a control theoretic perspective, they dictate the behavior of the system, and even slight changes can result in different systems. We can see that for 8 human subjects, the eigenvalues are almost identical. This finding agrees with neuroscientific results on different experimental settings [57], further demonstrating GEBM’s ability to provide useful insights on multi-subject experiments. For subject #3 there is a deviation on the real part of the first eigenvalue, as well as a slightly deviating pattern on the imaginary parts of its eigenvalues. Figures 3.4(c) & (d) compare matrix $A$ for subjects 1 and 3. Subject 3 negative value on the diagonal (blue square at the (8, 8) entry), a fact unique to this specific person’s connectivity.

Moreover, according to the person responsible for the data collection of Subject #3:

> There was a big demonstration outside the UPMC building during the scan, and I remember the subject complaining during one of the breaks that he could hear the crowd shouting through the walls.

This is a plausible explanation for the deviation of GEBM for Subject #3.

3.2 App2: Knowledge Base

| Main Goal: | Given a large Knowledge Base (KB), find synonyms of noun-phrases already in the KB, as well as “complete” the KB by inferring true facts about the world that the KB is missing. |

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...
Figure 3.4: **Multi-subject analysis: Subject #3 is an anomaly.** Sub-figures (a) and (b), show the real and imaginary parts of the eigenvalues of matrix $A$ for each subject. For all subjects but one (subject #3) the eigenvalues are almost identical, implying that the GEBM that captures their brain activity behaves more or less in the same way. Subject #3 on the other hand is an outlier; indeed, during the experiment, the subject complained that he was able to hear a demonstration happening outside of the laboratory, rendering the experimental task assigned to the subject more difficult than it was supposed to be. Sub-figures (c) and (d) show matrices $A$ for subject #1 and #3. Subject #3’s matrix seems sparser and most importantly, we can see that there is a negative entry on the diagonal, a fact unique to subject #3.

Figure 3.5: 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.

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 $A, B, C$. The columns of these low-rank embeddings can serve as soft-clustering indicators, for semantically similar triplets of (subjects, verbs, objects). Figure 3.5 shows a pictorial, fictitious example of a (subject, verb, object) KB tensor and its sparse PARAFAC decomposition: We can see that the first component has subjects like “Obama” and “Merkel”, objects like “USA” and “Germany”, and verbs such as “is president of” and “is chancellor of”; this indicates that the first rank-one component corresponds to leaders. Accordingly, by inspecting the non-zero values of the $F$-th component in Fig. 3.5, we can conclude that the component
corresponds to cars and racing.

In our completed work, we have focused on a slightly more general case than the (subject, verb, object) tensor, where we have (noun-phrase, noun-phrase, context phrase) coming from the NELL project. In Fig. 3.6(a) we show an example of latent groups as extracted from a $24M \times 24M \times 48M$ tensor using GIGATENSOR [59].

Furthermore, suppose that given this KB tensor, and given a particular noun-phrase, we would like to find similar (contextually) noun-phrases (i.e. noun-phrases that are frequently used in the similar contexts as the given noun-phrase). In order to do that, we can look at matrix $A$ which is a low rank embedding of all noun-phrases. In particular we can take $AA^T$, and thus, by looking at the highest scoring coefficients for the given noun-phrase, we can discover contextually similar noun-phrases. An example of such analysis, conducted using PARCUBE in [41] on a smaller (however still very large) dataset from NELL of size $14545 \times 14545 \times 28818$, is shown in Fig. 3.6(b).

We have done preliminary work on Knowledge Base mining using tensors in [59],[41], however, as we point out in the proposed work, there is still a lot to be done.

<table>
<thead>
<tr>
<th>Noun Phrase 1</th>
<th>Noun Phrase 2</th>
<th>Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>internet</td>
<td>protocol</td>
<td>'np1' 'stream' 'np2'</td>
</tr>
<tr>
<td>file</td>
<td>software</td>
<td>'np1' 'marketing' 'np2'</td>
</tr>
<tr>
<td>data</td>
<td>suite</td>
<td>'np1' 'dating' 'np2'</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Concept: “Web Protocol”</th>
</tr>
</thead>
<tbody>
<tr>
<td>internet</td>
</tr>
<tr>
<td>file</td>
</tr>
<tr>
<td>data</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Concept: “Credit Cards”</th>
</tr>
</thead>
<tbody>
<tr>
<td>credit</td>
</tr>
<tr>
<td>Credit</td>
</tr>
<tr>
<td>library</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Concept: “Health System”</th>
</tr>
</thead>
<tbody>
<tr>
<td>health</td>
</tr>
<tr>
<td>child</td>
</tr>
<tr>
<td>home</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Concept: “Family Life”</th>
</tr>
</thead>
<tbody>
<tr>
<td>life</td>
</tr>
<tr>
<td>family</td>
</tr>
<tr>
<td>body</td>
</tr>
</tbody>
</table>

(a) Latent concepts in NELL [59]  (b) Contextually similar noun-phrases [41]

Figure 3.6: Preliminary results on the NELL KB data

3.3 **App3: Multi-Aspect Social Networks**

**Main Goal:** Given multiple views of a social network, as well as spatio-temporal, demographic, and textual (e.g. posts that they write) for its users information, identify communities (and correlate them with the additional dimensions), as well as normal and abnormal activity in the network.

We distinguish the completed work on Multi-Aspect Social Networks in two parts: Multi-View Social Networks (§3.3.1) and Time-Evolving Social Networks (§3.3.2).
3.3.1 Multi-View Social Networks

Problem Definition:

Given: Multiple alternative views of a social network (who-calls-whom, who-messages-whom) given as a large and sparse (person, person, view) tensor.

Find: A sparse decomposition of the tensor

To: Detect communities within the network, that are consistent across all views.

Consider multi-aspect measurements of a social network; different aspects can be time or different views of the network. In [45] we introduce GRAPHFUSE, a tensor based Multi-View Social Network community detection algorithm, and 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 [45], we also provide a data mining case study on the REALITYMINING dataset. This dataset was introduced in [28] 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.

![Figure 3.7: Results on the four views of the REALITYMINING multi-graph. Red dashed lines outline the clustering found by GRAPHFUSE [45].](image)

In Fig.3.7, we show all four views of the dataset as clustered 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.

3.3.2 Time-Evolving Social Networks

Problem Definition:

Given: A time evolving social network given as a large and sparse (person, person, time) tensor.

Find: A sparse decomposition of the tensor

To: Detect groups/communities within the network along with their communication profile over time, and potential anomalies.

Whenever we have time as an additional aspect of a social network, our analysis is able to incorporate
it in the results, by offering temporal profiles of the latent communities that exist in the network. These temporal profiles are crucial for understanding the dynamics of the network, correlating the activities of the communities with external events (see ENRON data in the next paragraph), or detecting suspicious and abnormal activity that might point to malicious member of the network.

In [41] we use PARCUBE to analyze ENRON. This very well known dataset contains records for 44 months (between 1998 and 2002) of the number of emails exchanged between the 184 employees of the company, forming a $184 \times 184 \times 44$ of 9838 non-zero entries. In Figure 3.8 we illustrate the temporal evolution of the 4 most prevailing groups in our analysis, having annotated the figure with important events in the history of the company’s downfall, corresponding to peaks in the communication activity. Labelling of the groups was done manually.

![Temporal evolution of ENRON groups](image)

**Figure 3.8:** Temporal evolution of 4 groups in the ENRON dataset. We have labelled the groups, according to the position of the participants in the company. The labels of the extracted groups are consistent with other works in the literature [13, 42].

In addition to ENRON, we also analyze a snapshot of FACEBOOK using PARCUBE; the particular dataset we work on first appeared in [61], and consists of triplets of the form (Wall owner, Poster, day), where the Poster created a post on the Wall owner’s Wall on the specified timestamp. By choosing daily granularity, we formed a $63891 \times 63890 \times 1847$ tensor, comprised of 737778 non-zero entries. In Figure 3.9 we present our most surprising findings: On the left subfigure, we demonstrate what appears to be the Wall owner’s birthday, since many posters posted on a single day on this person’s Wall; this event may well be characterized as an ”anomaly”. On the right subfigure, we demonstrate what ”normal” FACEBOOK
activity looks like.

Finally, in [9] we discover temporal communities on social/phonecall time evolving networks, using efficient rank-one tensor decompositions. It is important to note that ignoring or aggregating the temporal aspect in all the above cases would have made discovery of such events much more difficult, if not impossible.
Chapter 4

Completed Work: Algorithms

This task is divided into two sub-tasks, Scalability & Speed (§4.1) and Interpretability (§4.2).

4.1 Alg1: Scalability & Speed

| Main Goal: Given very large (and usually very sparse) tensor and coupled data, compute their decom-
| position in a fast, efficient, and scalable manner. |

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 [59] we develop the first scalable algorithm for tensor decompositions on Map/Reduce; at the time of publication, [59] was able to decompose problems larger by at least two orders of magnitude than the state of the art. Subsequently, in [16], we developed a Distributed Stochastic Gradient Descent method for Map/Reduce that is able to scale to billions of parameters.

Not necessarily being restricted to the Map/Reduce framework, in [41] we propose PARCUBE, a novel, approximate, parallelizable algorithm for tensor decomposition which is able to analyze very big tensors on a single workstation.

The main idea behind PARCUBE is the following

1. Get a biased sample of indices of $X$ 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 4.1. 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. 4.1 that indices that were not sampled, are shown in white in the final result, indicating that they are exactly equal to 0.

In [41] 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
Figure 4.1: The main idea behind PARCube [41]: 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.

though the tensor itself might not fit in memory, we can choose the sub-tensors appropriately so that fit in memory, and we can compensate by extracting many independent sub-tensors. Figure 4.2 shows the relative approximation error of PARCube and PARAFAC, while increasing the number of repetitions (i.e. extracting different sampled tensors and decomposing them in parallel). We observe that for a small number of repetitions, there is an understandable gap between the ideal approximation error of PARAFAC, but as we run more repetitions, we explore the data more effectively, converging to the same approximation error (indicated by fact that the relative objective function cost on Fig. 4.2 converges to 1).

Figure 4.2: **PARCube approximates the PARAFAC accuracy.** The vertical axis shows the relative error (whose best value is 1) and the horizontal axis shows the number of repetitions of the sampling process. We see that PARCube approximates relative error of 1.

In [47], we extend the idea of [41], 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, as shown in Fig. 4.3. Our work in [41] and [47], even though backed up with theoretical guarantees for correctness of the approximation, is largely heuristic based; however, in [49], we formally define PARACOMP, a parallel algorithmic framework, similar to the spirit of spawning different
decompositions and merging the results, that guarantees identifiability of the latent factors.

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. 4.1, each of the smaller sampled pieces of the data can be decomposed by any solver, as long as this solver guarantees a locally optimal solution to the problem (which is the standard for tensor and coupled decompositions).

![Figure 4.3: TURBO-SMT is up to 200x faster, for comparable accuracy.](image)

An important aspect of interpretability in tensor decompositions is sparsity of the latent factors. Previously [42, 43] PARAFAC-SLF has been introduced, which is a PARAFAC model with $\ell_1$ norm penalties which leads to sparse latent factors; additionally we also show that, when the latent factors of PARAFAC have only a few non-zeros, one can perform higher order co-clustering.

Besides building faster and more scalable algorithms for already seasoned and well established Multi-Aspect/Tensor data analysis models, we have also worked on the development of new, intuitive and interpretable models.

An important aspect of interpretability in tensor decompositions is sparsity of the latent factors. Previously [42, 43] PARAFAC-SLF has been introduced, which is a PARAFAC model with $\ell_1$ norm penalties which leads to sparse latent factors; additionally we also show that, when the latent factors of PARAFAC have only a few non-zeros, one can perform higher order co-clustering.

Besides the main advantage of speedup and parallelization, PARCUBE [41] and TURBO-SMT [47], are able to produce sparse latent factors for very large tensors, where PARAFAC-SLF is very hard to scale. The reason why this happens is sampling. In both PARCUBE and TURBO-SMT we sample indices of rows, columns and fibers of the data, and extract a subtensor that is addressed by those indices. As we mentioned previously, the sampled indices will be orders of magnitude fewer than the full range of indices that span the tensor. As a result, when we return to merge the partial results from the samples, we will output a value only for those indices that were sampled. This leaves a very large portion of indices (that...
were not sampled) being zero at the final result, thus leading to sparse latent factors.

One might argue that we could have, instead, pre-processed the data and removed those indices before the decomposition, however, PARCUBE and TURBO-SMT allow for the flexibility of refining the result by extracting more sample tensors from the data, effectively exploring the space of the data while not adhering to a static pre-processing. To corroborate the fact that sparsity obtained through this process is indeed a meaningful and compact representation of the data, in Fig. 4.4 we compare the size of the output factors of PARCUBE and PARAFAC-SLF (which is the state of the art when it comes to latent factor sparsity), as a function of the number of samples (repetitions). PARCUBE starts 4 times sparser than PARAFAC-SLF (which might indicate that some important dimensions are left out) and as we run more repetitions of the sampling, PARCUBE produces factors as sparse as PARAFAC-SLF. In extensive experiments we conducted in [41], we observed that the results of PARCUBE are more than 90% sparser than the ones from PARAFAC, while maintaining the same approximation error.

![Figure 4.4: PARCUBE outputs sparse factors](image)

Figure 4.4: **PARCUBE outputs sparse factors**: Relative Output size (PARCUBE/ PARAFAC-SLF) vs sampling factor $s$ (where no. of repetitions is $r = 2s$. “lambda” is the $\ell_1$ norm penalty for PARAFAC-SLF.
Chapter 5

On-going & Proposed Work

As in our completed work, proposed work is divided in Applications (§5.1) and Algorithms (§5.2).

5.1 Applications

In this section we outline the proposed work for Application tasks App1 (§5.1.1), App2 (§5.1.2), and App3 (§5.1.3.1 & §5.1.3.2).

5.1.1 App1: Modelling the Functional Connectivity of the Brain

High Level Research Questions:

• Can we apply models such as the Recurrent Neural Network in modelling the functional connectivity of the brain?

We are particularly interested in a specific variant of RNNs, called Echo State Networks (ESNs) [32] In ESNs, we have the mapping of an input signal into a set of neurons that comprise the dynamical reservoir. There can exist connections between any of these neurons, or reservoir states. The output of the reservoir, at the last step, is transformed into the output signal, which is also referred to as teacher signal. The parallelism between ESNs and GE BM is as follows: if we consider the reservoir states to correspond to the latent neuron regions of GE BM, then the two models look conceptually very similar, in this high level of abstraction. Additionally, in a similar experimental setting, Wehbe et al. [62] recently showed promising results towards this direction.

To further corroborate to the connection of our proposed model GE BM to ESNs, usually the connections between reservoir states are desired to be sparse, which is reminiscent of GE BM’s specification for $A$ to be sparse. In order to formalize the connection between GE BM and ESNs, here we provide the system equations that govern the behavior of an ESN [32]

\[
x(t + 1) = f \left( Wx(t) + W^{in}u(t + 1) + W^{fb}y(t) \right)
\]

\[
y(t) = g \left( W^{out} [x(t) \ u(t)] \right)
\]

where $f$ and $g$ are typically sigmoid functions. Vector $x(t)$ is the so-called reservoir state, $W$ is the connectivity between the reservoir states, $W^{in}$ is the input transformation matrix, $W^{fb}$ is a feedback matrix, and $W^{out}$ transforms the hidden reservoir states to the observed output signal of vector $y(t)$. If
we set $f$ and $g$ to be identity, set $\mathbf{W}^{fb} = \mathbf{0}$, and set the part of $\mathbf{W}^{out}$ that multiplied $\mathbf{u}(t)$ to be zero as well, then the above ESN equations correspond to GEBM.

### 5.1.2 App2: Robust Knowledge Base Completion & Synonym Discovery

<table>
<thead>
<tr>
<th><strong>High Level Research Questions:</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>In the context of modelling a KB as a tensor:</td>
</tr>
<tr>
<td>• How can we decide whether we should complete a missing value in the KB or not?</td>
</tr>
<tr>
<td>• Can we incorporate type information in KB completion?</td>
</tr>
</tbody>
</table>

A Knowledge Base (KB) contains triplets about facts that it already “knows” about the world. However, the majority of potential triplets are missing. Triplets that are missing from the KB 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. In particular, we propose to pursue the follow directions:

1. **Selective Imputation:** Intuitively, most of the missing / unobserved values of the KB should be implausible, and only a few ought to be automatically completed. This sparsity assumption on the set of the values that can be completed poses very strong prior information that we intend to concretely model and incorporate in tensor completion approaches, in order to impute only a few, but highly likely to be plausible, missing entries of the KB.

2. **Using Type Information:** A separate interesting idea, as demonstrated by [22] is the restriction of the decomposition by types of the entities and their relations. For instance, a type may explicitly declare that a car is not edible, and therefore, this constraint would further restrict the space of potential imputations. Here, we propose to investigate how we can model types as side information in the form of a coupled matrix for the KB tensor.

### 5.1.3 App3: Multi-Aspect Social Networks

In Task App3 we have two proposed sub-tasks: Location Based Social Networks (§5.1.3.1) and Analyzing Bilingual Immigrant Communities (§5.1.3.2).

#### 5.1.3.1 Location Based Social Networks

<table>
<thead>
<tr>
<th><strong>High Level Research Questions:</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Can we detect normal and abnormal behavior from social data with location information?</td>
</tr>
<tr>
<td>• Can we use location information to improve friend recommendation, and friendship information to improve venue recommendation?</td>
</tr>
</tbody>
</table>

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 [41, 38], 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 [44].

5.1.3.2 Analyzing Bilingual Immigrant Communities

<table>
<thead>
<tr>
<th>High Level Research Questions:</th>
</tr>
</thead>
<tbody>
<tr>
<td>In the context of modelling the above immigrant web community as a tensor:</td>
</tr>
<tr>
<td>• Can we automatically identify bilingual users in immigrant communities?</td>
</tr>
<tr>
<td>• What are the topics where bilingual users code-switch the most?</td>
</tr>
<tr>
<td>• Are users strictly bilingual or monolingual, or does it depend on factors such as the discussion topic, the social context, or the time?</td>
</tr>
</tbody>
</table>

In this particular application, we are interested in discovering communities of bilinguals. A person is considered bilingual if she is fluent in two languages; this trait is very common among immigrant communities. Among such bilingual individuals, it is very common to code-switch, i.e. to switch between two different languages during the same sentence or paragraph. The particular data we are proposing to study come from a very large online forum in the Netherlands that is used by the Turkish immigrant community there. A preliminary study and description of the data can be found in [48]. The posts are mostly in Dutch and Turkish, and the forum is divided into several sub-fora, each one specific to a particular topic. Empirically, bilingual users there code-switch frequently, either within the same post, or across posts. We propose to model the above forum as a very high dimensional tensor, including the users, the topics or sub-fora, and the words they use when posting on specific sub-fora. Our aim is to identify coherent communities of users that code-switch, topics of sub-fora that are mostly bilingual, and topics that are mostly monilingual.

5.2 Algorithms

Our proposed work for the Algorithms task mainly fall under Interpretability (task Alg2). However, everything outlined in the following lines implies that the algorithms we propose to develop will have to be efficient, fast, and scalable, thus Scalability (task Alg1) is an overarching theme.

5.2.1 Alg2: Interpretability

The first proposed task for Interpretability is mostly concerned with evaluating how well our algorithms can model a particular dataset, as well as how many components are hidden in the data, especially when ground truth is not available. The second task proposes to tackle cases where data can be structurally problematic, leading already existing algorithms to instabilities and degenerate solutions.
### 5.2.1.1 Unsupervised Quality Assessment

<table>
<thead>
<tr>
<th>High Level Research Questions:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Given a large and potentially very sparse tensor, and its PARAFAC decomposition, is the decomposition meaningful?</td>
</tr>
<tr>
<td>• Given a large and potentially very sparse tensor, what is a good number of components for its PARAFAC decomposition?</td>
</tr>
</tbody>
</table>

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. In particular, given, say a tensor $\mathbf{X}$ and its $F$-component decomposition $[\mathbf{A}, \mathbf{B}, \mathbf{C}]$, we would like to have with a robust and scalable method that reflects how well do $[\mathbf{A}, \mathbf{B}, \mathbf{C}]$ capture the structure of the data in $\mathbf{X}$.

There exist heuristics in the literature, such as CORCONDIA [20] 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), as well as for different loss functions, such as KL-divergence, which is shown to be more appropriate when we have sparse count data [24]. 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.

### 5.2.1.2 Mode Redundancy & Structural Abnormalities

<table>
<thead>
<tr>
<th>High Level Research Question:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Given structurally problematic data, can we derive robust methods that are able to extract useful information despite these imperfections?</td>
</tr>
</tbody>
</table>

In addition to quality assessment proposed in the previous task, we would also like to have robust algorithms that are able to automatically filter out aspects of the data that are noisy or redundant. The following exemplary scenarios draw the outline of the behavior that we would like our algorithms to be able to counteract.

1. Consider a brain measurement tensor (like the one in [47]) with modes: (nouns, voxels, human subjects). If the responses of all human subjects are very similar (for the sake of argument every slice is a copy of the exact same matrix), then the tensor will look like Fig. 5.1(a). Thus, the Third mode is redundant since it simply replicates the same information. How can we automatically detect that and avoid trying to interpret the redundant mode?

2. Consider the case of a three mode KB tensor where the first two modes are subject and object and the third mode is the verb. Suppose now that the possible values for verb are “leads”, and “governs”. Say our subjects are “Obama”, “Merkel”, and “Cameron” and the objects are “USA”, “Germany”, and “UK”. The only valid triples that exist connect “Obama”, using all the verbs, with “USA”, “Merkel” with “Germany”, and “Cameron” with “UK”, but there cannot be a triple that connects, “Obama” with, say, “Germany”. Thus, in this problematic case, the tensor will look like the one shown in Fig. 5.1(b) and does not resemble the block structure that tensor decompositions seek to extract.

The two cases we mention are two extreme, fictitious examples. In reality, a dataset might suffer from a
Figure 5.1: Two extreme problematic cases, as described in the text above. In reality, a tensor might have parts of it with such structural problems, and other parts with “good” structure.

Combination of these problems, for a subset of its dimension, but for a different subset, the data might have useful and exploitable structure. In said cases, we have issues where traditional decomposition models and algorithms can possibly produce a result, however the result will be highly unstable and thus the analysis may be of questionable quality. We propose to examine the above cases and derive models and algorithms that are able to treat the above problems in a robust and, ideally, problem independent way.
Chapter 6

Related Work

6.1 Tensor Decompositions and Applications

There exist a rich literature of different tensor decompositions. In this work we are mostly using and referring to the PARAFAC decomposition, independently introduced in [30] (as PARAFAC) and in [21] (as Canonical Decomposition). The PARAFAC decomposition, as also described in Sec. 2 is suitable for uncovering the latent factors of the tensor. More recently bayesian/probabilistic versions of the decomposition have been introduced [64]. However, other decompositions such as TUCKER3, introduced in [58] are more suitable for efficiently compressing tensors. A very thorough survey on all the existing tensor decompositions can be found in [35]. In terms of state of the art toolboxes, the N-Way Toolbox for Matlab [8] specializes in dense tensors, while the Tensor Toolbox for Matlab [14] is highly optimized for sparse tensors.

Tensor Applications to Brain Data

There has been substantial related work, which utilizes tensors for analyzing brain data. In [2] Acar et al. apply the PARAFAC decomposition on EEG measurements in order to detect epileptic seizures. Morup et al. in [40] introduce a variation of the PARAFAC decomposition that is able to handle temporal shifts across brain measurements. In [25], the authors describe how one can use non-negative tensor factorization for source separation and analysis of brain signals, and finally, in [26] et al. introduce a constrained PARAFAC model in order to identify the connectivity between different brain regions.

Tensor Applications to Social Network Analysis

Social network analysis has also been tackled using tensors before. One of the first works that did so was the work of Bader et al. [10], where the authors detect cliques of dense interaction in the ENRON e-mail network, while tracking their activity over time. The findings of [10] have also been reproduced in [43] using different, albeit in the tensor regime, techniques. Lin et al. [36] employs coupled tensor factorization for community detection in social graphs with context information about the users. Finally, in [41], the authors apply their tensor decomposition algorithm on a Facebook dataset, detecting normal and abnormal user interaction.

Other Applications

Tensors and tensor decompositions have gained increasing popularity in the last few years, in the data mining community [35]. The list of tensor applications in data mining is long, however we single out a few
that we deemed representative: In [34], the authors extend the well known link analysis algorithm HITS, incorporating textual/topical information. In [13] and [12] the authors use tensors for social network analysis on the ENRON dataset. In [55], the authors propose a sampling-based TUCKER3 decomposition in order to perform content based network analysis and visualization. The list continues, including applications such as Cross-language Information Retrieval [23], Anomaly Detection [39], Brain Signal Analysis and detection of Epilepsy [2], Machine Vision [60], and Web Search [56] to name a few. In [31] the authors use tensor decompositions to infer Hidden Markov Models. Apart from Data Mining, tensors have been and are still being applied in a multitude of fields such as Chemometrics [18] and Signal Processing [50].

6.2 Coupled, Multi-block, Multi-set Models

Coupled Matrix-Tensor Factorizations belong to a family of models also referred to as Multi-block or Multi-set in the literature. Smilde et al. in [53] provided the first disciplined treatment of such multi-block models, in a chemometrics context. One of the earliest works that introduce the concept of coupling in data mining applications is [15], where the authors apply their algorithms to movie recommendation and newsgroup article clustering. In [52], Singh and Gordon introduce a collective matrix factorization framework, again in a data mining setting, where the coupling is between matrices. An important issue with these models is how to weigh the different data blocks such that scaling differences may be alleviated. In [63], Wilderjans et al. propose and compare two different weighing schemes.

Most related to the present work is the work of Acar et al. in [4], where a first order optimization approach is proposed, in order to solve the CMTF problem. In [5], Acar et al. apply the CMTF model, using the aforementioned first-order approach in a bioinformatics setting. In [3], Acar et al. introduce a coupled matrix decomposition, where two matrices match on one of the two dimensions, and are decomposed in the same spirit as in CMTF, while imposing explicit sparsity constraints (via $\ell_1$ norm penalties). As an interesting application, in [67], the authors employ CMTF for Collaborative Filtering. On a related note, [65], [36], and [37] introduce models where multiple tensors are coupled with respect to one mode, and analyzed jointly. Finally, extending the CMTF framework, Acar et al. propose in [7] a new model where rank-one components can be shared by the matrix and the tensor, or belong solely to either one of the data pieces/blocks.

6.3 Fast & Scalable Tensor Decompositions

The mechanics behind the Tensor Toolbox for Matlab [14] are introduced in [34] and later on in [11], where Bader and Kolda first show how to efficiently decompose sparse tensors, avoiding the materialization of prohibitively dense and large intermediate data. Bro and Sidiropoulos introduce a compression based method in [19] which is able to speed up PARAFAC, without however exploiting parallelization. In [41] we introduced a parallel algorithm for the regular PARAFAC decomposition, where a sampling scheme of similar nature as here is exploited; following a similar split and merge parallelization scheme, [51] introduces a fast and parallelizable algorithm for PARAFAC, based on compression using random projections, which provides guarantees with respect to the identifiability of the true latent factors. In [59], a scalable MapReduce implementation of PARAFAC is presented, where the intermediate data problem is avoided in the same way as in [34, 11], with specific considerations that pertain to the distributed implementation. In [66], the authors introduce a parallel framework in order to handle tensor decompositions efficiently. More recently, Beutel et al. [17] introduce a highly scalable Distributed Stochastic Gradient system for CMTF on MapReduce. In [27], the authors propose an alternative parallel architecture for decomposing tensors on a cluster, by exploiting the multilinear nature of the tensor data and mapping the computation in
a similar multilinear fashion on the cluster’s machines. Finally, in [33] the authors introduce a method to vertically partition a multi-relational dataset (which can be viewed as a multiway tensor) using relational normalization, which results in independent decompositions of smaller tensors, that are combined at the end.
Chapter 7

Timeline

My expected timeline goes as follows:

- **February 13, 2015** - Thesis proposal
- **February 2015 - May 2015** - Analyzing Bilingual Immigrant Communities
- **February 2015 - May 2015** - Unsupervised Quality Assessment
- **February 2015 - May 2015** - Robust KB Completion
- **April 2015 - May 2015** - Location Based Social Networks
- **April 2015 - June 2015** - Mode Redundancy & Structural Abnormalities
- **June 2015 - August 2015** - Summer internship
- **September 2015 - November 2015** - Modelling the Functional Connectivity of the Brain
- **November 2015 - December 2015** - Thesis writing
- **December 2015** - Job applications
- **February 2016 - March 2016** - Interviewing
- **April 2015 - May 2015** - Thesis writing
- **May 2016** - Thesis defense
Chapter 8

Conclusion

In this thesis, we propose to tackle data mining problems for large data that occur in multiple aspects. In order to do so, we work in two thrusts:

- **Algorithms:** We focus mainly on tensors and tensor decomposition, and we develop fast, scalable and interpretable algorithms that are able to tackle a rich variety of multi-aspect data mining applications.

- **Applications:** We apply our algorithms to three major applications:
  - Neurosemantics: Extracting useful information and predictive models for the human brain, when a person is reading, in order to enhance our understanding of the human brain.
  - Knowledge Base (KB): Discovering synonyms for noun-phrases in a vast KB, and enriching the KB with new facts.

As we’ve described in Chapter 3 our completed work has already addressed a considerable variety of tasks within the three major applications. Furthermore, with respect to Algorithms (Chapter 4), our progress is highlighted by methods that are up to two orders of magnitude faster than the state of the art, while maintaining accuracy comparable or equal to the state of the art and results up to 90% sparser. Finally, we promote reproducibility and re-usability of our results and algorithms by making our code publicly available at [http://www.cs.cmu.edu/~epapalex/code.html](http://www.cs.cmu.edu/~epapalex/code.html).
Bibliography


