Mining Interesting Local Structure from Tensors

Abhinav Maurya
Information Systems
Carnegie Mellon University
ahmaurya@cmu.edu

Manzil Zaheer
Machine Learning
Carnegie Mellon University
manzil@cmu.edu

November 7, 2014

Abstract

Tensors are ubiquitous in real-world recommender systems. In this project, we focus on finding interesting local structure in massive real-world tensors beyond tensor decompositions. For example, in a tensor of $subject - relation - object$ triplets derived from the NELL dataset, we consider the question of finding the most interesting and important facts about a *subject* or a *relation*. We suggest *CF-ICF* – a variant of TF-IDF – to answer this question. Another question that we consider is how to discover and characterize dense and semantically coherent local blocks in the tensor. To answer this question, we propose *Bayesian Tensor Blockmodels* as a method that detects and characterizes dense tensor blocks. We perform a literature survey of important related work in the area, and evaluate our solutions on the tensors constructed from the Yago and NELL datasets.

1 Introduction

Tensors occur in many real-world datasets. For example, in the NELL dataset, the three-mode tensor consists of $subject - verb - object$ pairs. In the Yago knowledge base, the tensor consists of $entity_1 - relation - entity_2$ triplets. We consider the problem of finding interesting facts given a particular value of a particular mode of the tensor. For example, what are the most interesting/surprising facts for *Obama* or *Carnegie Mellon University*? A primary question that we need to answer in this context is to define the level of interestingness of a fact. How do we quantify this? Is it based on its surprise factor to humans? Is it based on its diversity from common, banal facts about the noun? How do we formally capture this notion? These are the interesting questions we seek to answer in this project. Another question that we consider is how to find semantically coherent tensor blocks from real-world tensors. To address this question, we propose *Bayesian Tensor Blockmodel* to find dense tensor blocks that are explained by semantically meaningful concepts along each mode of the tensor block.

Outline: The rest of the paper is organized as follows. We begin by formally describing the problem statement in Section 2 and provide an in-depth literature survey in Section 3. Next we describe the datasets which we would to evaluate all of our methods in Section 4. Then we start addressing the goals of the paper, one by one, beginning with finding interesting facts in Section 5, finding important facts in Section 6 and finding semantically coherent facts in Section 7. In all of the proposed approaches we demonstrate that efficacy through carrying out experiments on the dataset described. Finally, we conclude in Section 8.

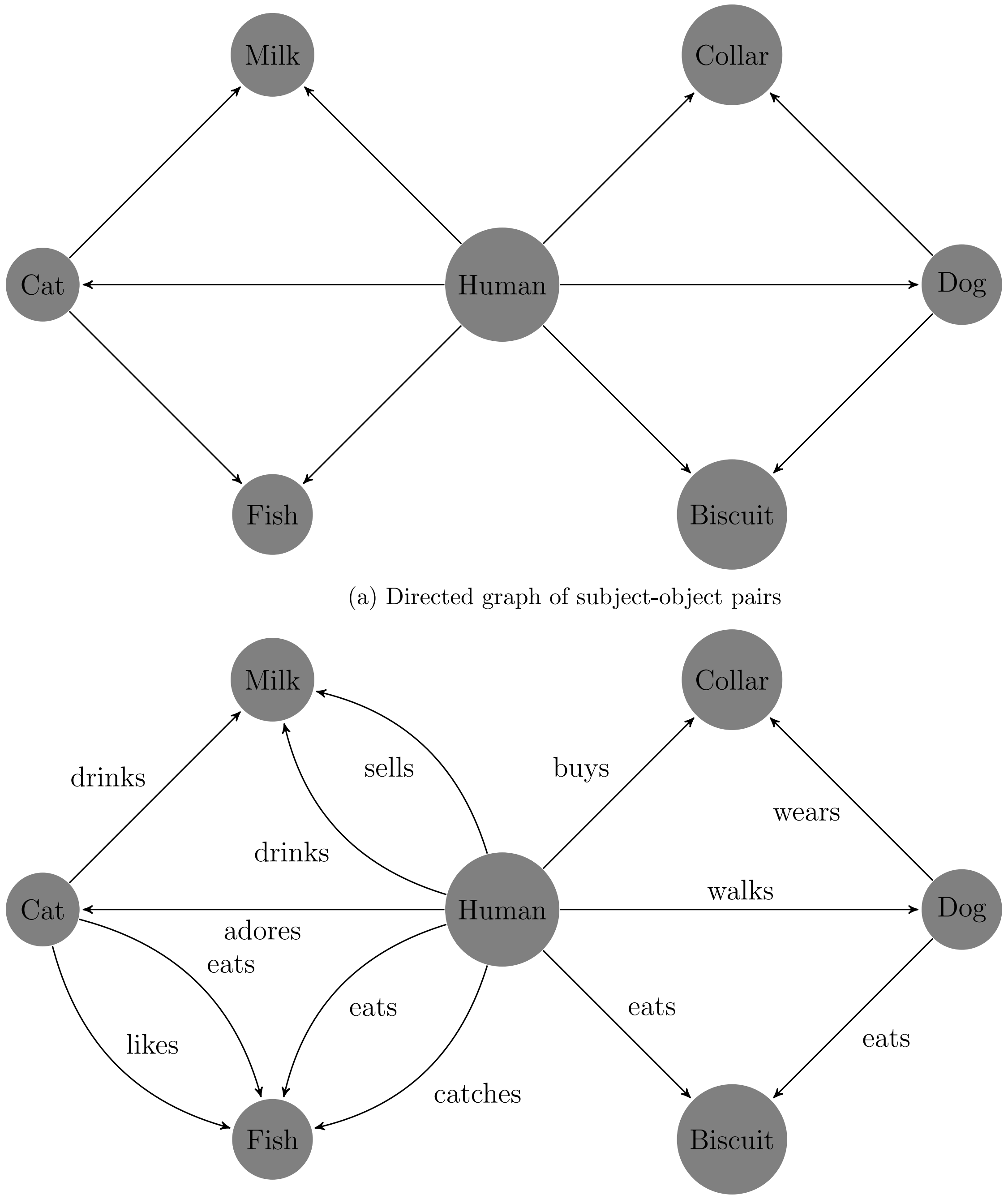
2 Problem Definition

We limit ourselves to multi-relational data in the form of triplets, where two entries belong to the same universe, e.g. in the subject-verb-object case, both subject and object come from the same universe of nouns. A scalar is denoted by lower-case italic letter or greek letter, e.g. d, α. A vector is denoted by a bold lower-case letter, e.g. $\mathbf{v}$ whose i^{th} entry is $\mathbf{v}(i)$. Matrices are denoted by bold upper-case letters, e.g. $\mathbf{W}$ with $(i,j)^{th}$ entry $W(i,j)$. Sets are denoted by upper-case calligraphic letters, e.g. $\mathcal{A}$. Tensors are denoted by upper case frakturs letters $\mathfrak{X}$ whose $(i,j,k)^{th}$ entry is $\mathfrak{X}(i,j,k)$. Table 1 gives a list of common symbols we used.

The main motivation behind our method is to find most interesting triplets from a given (possibly huge) set of facts as subject-verb-object triplets. Here both subject and object belong to the same universe. If instead of triplets, suppose we were only provided with a list of subject-object pairs $\mathcal{L} = \{(a_1, b_1), ..., (a_l, b_l)\}$. Then it can be represented as a directed graph $G = (V, E)$ with set of vertices $V = \mathcal{S} = \texttt{noun}$ and edges $E = \mathcal{L}$ as shown in figure 1(a). Then most popular vertex can be found using the PageRank algorithm.

But now we have a list of subject-verb-object triplets $\mathcal{L} = \{(a_1, c_1, b_1), ..., (a_l, c_l, b_l)\}$.The triplets can be viewed as a labelled multi-graph $G = (V, E, A)$ with set of vertices $V = \mathcal{S} = \texttt{noun}$ and edges $E = \mathcal{L}$ having labels from set $\mathcal{A} = \texttt{verbs}$ shown in figure 1(b). Finding popular and interesting facts now becomes a challenging and open problem, which we aim to target. We also address the question of finding dense tensor blocks in a given tensor that can be coherently explained by topics built on words used to describe dimensions of each mode of the tensor.

Also a big question, is what do we precisely mean by interesting fact? Unlike popularity



(a) Directed graph of subject-object pairs

(b) Labelled directed multi-graph of subject-verb-object triplets

Figure 1: Representative graphical representation

Symbol	Definition
$\mathfrak{X}$	Tensor containing data
$\mathcal{T}$	Tensor containing transition probabilities
$\mathcal{S}$	Set of vertices/nouns
$\mathcal{A}$	Set of relations/verbs
n	Number of vertices/nouns
m	Number of relations/verbs
l	Number of entries in the tensor
$\mathfrak{X}_{(k)}$	mode-k matricization of a tensor
$\circ$	outer product
$\odot$	Khatri-Rao product
$*$	Hadamard Product

Table 1: Symbols and definitions

or importance which have been well studied, the interestingness is much more complicated and not studied yet much to the best of our knowledge. To elaborate, spotting obscure facts about an obscure entity is not very interesting, e.g. "Riemann zeta function obeys analytic continuation". However, to humans it is very interesting in spotting an obscure fact about a popular entity. For example, everybody would be interested to know that "Michael Jordan (a famous basketball player) is afraid of water". Thus, *specificity* in a fact is strong indicator of surprise factor present in it. So we wish to mine for such facts from the tensor.

3 Survey

Next we list the papers that each member read, along with their summary and critique.

3.1 Papers read by Abhinav Maurya

3.1.1 Local Low-Rank Matrix Approximation [4]

- *Main idea*: The paper relaxes the low-rank assumption often made in matrix decomposition. Instead, the assumption is that the matrix is a weighted combination of matrices that are low-rank in specific neighborhoods of the matrix indices. This assumption is less restrictive than a low-rank assumption which forces the same latent factors to be shared throughout all neighborhoos of indices. Low-rank matrix approximation is done using two methods:

- Incomplete SVD: The matrix $\mathbf{X}$ is usually only sparsely observed and is therefore incomplete. This method constructs a low-rank approximation $\mathbf{X}'$ by minimizing its error with $\mathbf{X}$ on the observed entries.
- Compressed Sensing: This method minimizes the nuclear norm of the approximation $\mathbf{X}$ (since nuclear norm is the convex surropagte of the rank of a matrix) and such that the Frobenius norm of the difference between $\mathbf{X}$ and $\mathbf{X}'$ projected only on the observed entries is less than the desired error ϵ.

The paper reformulates these two ways of matrix completion for local low-rank matrix completion by choosing q local models $\hat{T}_1, \hat{T}_2, .., \hat{T}_q$ and combining these local models using the Watson-Nadaraya locally constant nonparametric regression to estimate the approximation $\hat{\hat{T}}$.

$$\hat{\hat{T}}(s) = \sum_{i=1}^{q} \frac{K_h(s_i, s)}{\sum_{j=1}^{q} K_h(s_j, s)} \hat{T}(s_i)$$

where $s, s_i, s_j \in [n_1] \times [n_2]$ i.e. they index into the matrix of dimensions $n_1 \times n_2$. Thus, the approximation at index s is the locally weighted average of local models $\hat{T}(s_i)$ at certain chosen indices s_i.

- *Use for our project*: We believe that tensors might have local low-rank structure just like matrices. Allowing local low-rank tensor decomposition might help us gain more accurate approximations for tensors.
- *Salient Contributions*: The paper relaxes the low-rank structure which might not be accurate for extremely large matrices. Instead, it introduces local low-rank assumption and finds better approximation methods for matrices that satisfy this assumption.
- *Shortcomings*: The paper assumes that there are q local low-rank models which are learnt by sampling q local neighborhoods from the observed entries of the matrix. Introducing bias in the sampling process may introduce the efficacy of the discovered q local models. Also, the combination of these local models using Watson-Nadaraya estimator instead of finding the combining weights using a principled optimization problem is unsatisfactory.

3.1.2 ParCube [7]

- *Main idea*: The central idea of ParCube is to sample a massive tensor such that the resultant sampled tensor can be used for decomposition instead of the original tensor. Instead of uniformly sampling from each mode, ParCube uses biased sampling, where the bias on each mode is dependent on the marginal sum of tensor entries on that mode. The computational complexity of the sampling algorithm is linear in the number of non-zero entries of the original tensor being sampled. Once we obtain the sampled tensor, any Parafac decomposition can be applied. After the decomposition, the factors are expanded out and redistributed to their original positions i.e. the values of indices that were not included in the sampling will be zero in the redistributed factors, inducing a natural sparsity in the reconstructed tensor approximation. The sampling of indices on each mode of the tensor is biased using the marginal sums across the mode. This does not take into account the distribution of values along that mode. If there are huge outlier, noisy values in the tensor, the sampling step will bias the sampling towards the indices of these outlier values. To counteract the sparsity enforced by sampling just once, the ParCube decompsosition algorithm maintains multiple sampled tensors and their decompositions and stitches these decompositions together using a common set of indices across all the samples.

- *Use for our project*: We propose to use NELL dataset as our main dataset. The massive dataset consists of a highly sparse 20M x 10 M * 20M tensor where the modes are subject, verb, and object. ParCube will help us decompose this tensor into highly sparse factors. These factors will be used to construct an appropriate reward function for an MDP that will find the most interesting verb and object for a given subject through a value iteration procedure.
- *Salient Contributions*: The sampling procedure makes it possible to factor massive tensors on commodity workstations, and stitch together multiple such tensor decompositions to get a sparse decomposition for the original tensor.
- *Salient Shortcomings*: The ParCube method does not enforce that the sparse set of indices be different across different factors of the decomposition. Indeed, in the final algorithm, a common set of indices across the various sampled tensors is used to stitch the factors across the decompositions together. Enforcing disjointness across the sparse set of indices of different factors of a tensor decomposition is useful because it helps us discover locally low-rank tensor structure, similar to recent work on locally low-rank matrix decompositions.

3.1.3 GigaTensor [2]

- *Main idea*: The paper describes extensions to ParaFac tensor implementation to execute it on a distributed Hadoop system. The extensions allow decomposition of massive sparse tensors that do not fit in the main memory of a single system. If $\mathfrak{X}$ is the tensor and $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are its factor matrices (the three vectors along the three modes in the single-rank tensor factor are stored in $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$), then the Alternating Least Squares (ALS) method involves the following step:

$$\mathbf{A} \leftarrow \mathfrak{X}_{(1)}(\mathbf{C} \odot \mathbf{B})(\mathbf{B}^T\mathbf{B} * \mathbf{C}^T\mathbf{C})^\dagger$$

The first optimization to use the associativity of matrix multiplication to multiply $\mathfrak{X}_{(1)}$ and $(\mathbf{C} \odot \mathbf{B})$ first. The second optimization is to not instantiate $(\mathbf{C} \odot \mathbf{B})$ and then multiply $\mathfrak{X}_{(1)}$ with it. Instead, an intelligent multiplication method is outlined in Algorithm 2 of the paper, to avoid the intermediate data explosion. Again to avoid data explosion, the outer products in $(\mathbf{B}^T\mathbf{B} * \mathbf{C}^T\mathbf{C})$ are calculated in a parallel fashion. The calculation of Moore-Penrose Pseudoinverse of $(\mathbf{B}^T\mathbf{B} * \mathbf{C}^T\mathbf{C})$ is easy since the matrix is small (i.e. $R \times R$). Finally the multiplication of $\mathfrak{X}_{(1)}(\mathbf{C} \odot \mathbf{B})$ and $(\mathbf{B}^T\mathbf{B} * \mathbf{C}^T\mathbf{C})^\dagger$ is done using distributed cache multiplication by broadcasting the smaller matrix $(\mathbf{B}^T\mathbf{B}*\mathbf{C}^T\mathbf{C})^\dagger$ to the mappers that process the larger matrix $\mathfrak{X}_{(1)}(\mathbf{C}\odot\mathbf{B})$ is a distributed fashion. GigaTensor makes possible the multiplication of tensors of sizes hundred times larger than those of tensors that could be decomposed earlier.

- *Use for our project*: The paper uses the NELL dataset, which we propose to use for our evaluation. We intend to use a tensor decomposition to discover concepts in the NELL dataset, and will require a scalable sparse tensor decomposition like GigaTensor in the first step of our goal of discovering the most interesting subject-verb-object triplets in the NELL dataset.
- *Salient Contributions* A careful analysis of the various operations involved in the ALS method of tensor decomposition leads to a highly distributed solution with major gains in speedups and the sizes of tensors that can be decomposed.
- *Salient Shortcomings*: The paper compares GigaTensor to the Tensor Toolbox which has not been parallelized. A comparison with a toolbox that utilizes parallelized matrix operations might have been more appropriate. Another option is to provide the huge matrices to be multiplied to a general distributed Hadoop matrix multiplier instead of hand-crafting a particular solution for the three-way tensor decomposition. Since the optimizations are tuned to three-way tensor decomposition, it is also not clear if the method performs well on the decomposition of general sparse n-way tensors.

3.2 Papers read by Manzil Zaheer

3.2.1 PageRank [1]

The first paper was on the PageRank by Brin and Page [1].

Main idea : PageRank algorithm is how Google got started ranking web pages. It involves random walks in directed graphs. PageRank considers a random walk that with probability d follows a random edge out of the present node, and with probability $1 - d$ jumps to a uniformly random vertex of a graph. The PageRank vector is the steady-state distribution of this process. That is we have a transition probability $\mathbb{P}\{X_{t+1} = i_2 | X_t = i_1\}$, which can be estimated as:

$$\mathbf{W}(i_1, i_2) = \frac{\mathbf{X}(i_1, i_2)}{\sum_{i_2=1}^{n} \mathbf{X}(i_1, i_2)} \tag{1}$$

where $\mathbf{X}$ is the adjacency matrix of the graph. Then PageRank vector $\mathbf{r}^*$ will satisfy:

$$\mathbf{r}^* = \frac{(1-d)}{n}\mathbf{1} + d\mathbf{W}\mathbf{r}^* \tag{2}$$

The vector r contains the score indicating importance of vertices (webpages).

Use for our project : Lays foundation of Random walks and its usefulness in finding most popular vertices

Shortcomings : Limited to simple graphs

3.2.2 Markov Decision Process [6]

The second paper was the book-chapter on Markov Decision Process by Mahadevan [6]. We describe this topic in detail, because we will need the details in describing the details of our proposed method.

A Markov decision process (MDP) is a tool to plan efficiently if the results of actions executed are uncertain. To be precise, the MDP is a 4-tuple $\Xi = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T})$, defined as:

- $\mathcal{S}$ is the set of states in the system and set $n = |\mathcal{S}|$. At every instant, the system must be one of the state $s \in \mathcal{S}$.

- $\mathcal{A}$ is set of actions and set $m = |\mathcal{A}|$. At each time step we can choose to take one of the action, i.e. carry out one of the action.

- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function.

- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Pi(\mathcal{S})$ is the conditional transition probability of moving to a new state $S_{t+1} \in \mathcal{S}$ given current state $S_t \in \mathcal{S}$ and action $A_t \in \mathcal{A}$. This can be represented as a tensor $\tau(s, a, s^+) = \mathbb{P}(S_{t+1} = s^+ | S_t = s, A_t = a)$.

Under assumptions of this model, the system evolves as follows: at each time point, the system is in a particular state, s, an action a is taken and there is a transition to another state s^+. However, we require that the state depend only upon s and a. In addition, s and a only give probabilistic information about what the resulting state will be according to.

Having described all the quantities of interest, the intention is to find a policy $\pi^* \in \mathcal{A}^{|\mathcal{S}|}$, specifying which algorithm to run so as to maximize our expected rewards, i.e.

$$\pi^* = \arg\max_{\pi \in \mathcal{A}^{|\mathcal{S}|}} \mathbf{v}_\pi \tag{3}$$

where
$$\mathbf{v}_\pi(S_0) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(S_t, \pi(S_t), S_{t+1})\right] \tag{4}$$

Now define $r(s, a) = \sum_{s^+ \in \mathcal{S}} \tau(s, a, s^+)\mathcal{R}(s, a, s^+)$ for simplicity. Now observe that following recurrence relation holds:

$$\mathbf{v}_\pi(s) = r(s, \pi(s)) + \gamma \sum_{s^+ \in \mathcal{S}} \tau(s, \pi(s), s^+)\mathbf{v}_\pi(s^+) \tag{5}$$

Then we can show that optimal reward value and policy is the fixed point solution of the celebrated Bellman equation:

$$\begin{aligned} \mathbf{v}^*(s) &= \max_{a \in \mathcal{A}} \left[r(s, a) + \gamma \sum_{s^+ \in \mathcal{S}} \tau(s, a, s^+)\mathbf{v}^*(s^+)\right] \\ \pi^*(s) &= \arg\max_{a \in \mathcal{A}} \left[r(s, a) + \gamma \sum_{s^+ \in \mathcal{S}} \tau(s, a, s^+)\mathbf{v}^*(s^+)\right] \end{aligned} \tag{6}$$

With abuse of notation, we can write it in vector format as:

$$\mathbf{v}^* = \max_{a \in \mathcal{A}} [r(:, a) + \gamma \mathcal{T}(:, a, :)\mathbf{v}^*] \tag{7}$$

We will use this MDP as new way to perform random walk to over the most “interesting” triplets as explained further in section 6.

3.2.3 Centrality Measure

I studied two papers about centrality measure. In the study of networks, a ”centrality” measure attempts to measure the importance of a node, an edge, or some other subgraph. As discussed before, PageRank provides a centrality measure on simple graphs. There have been attempts to generalize this or come up with different centrality measure for multi-graphs. We describe two of such methods, relevant to our project:

TOPHITS [3]

- *Main idea*: This paper provides modified version of the HITS algorithm. Basically HITS (hyperlink-induced topic search) views Web as a graph and provides a centrality measure composed of hub and authority scores for every vertex using a singular value decomposition (SVD) of the adjacency matrix of the graph. In TOPHITS, a third dimension representing anchor-text is added, which is claimed to be useful for web searches because"it behaves as a consensus title". Then, similar to HITS, they perform the tensor-decomposition method PARAFAC (which is a generalization of the matrix SVD) to obtain a rank-R approximation of the tensor. These PARAFAC decomposition tries to explain the data using the R topics. The singular values provide weight of the topic and the singular vectors provide the hub, authority and term score for each vertex for that topic. Now using these scores, they specify methods (simple linear operations) to answer two types of queries in a ranked fashion: find all pages related to a given term and find all pages related to a given webpage.

- *Use for our project*: To solve the PARAFAC decomposition an alternating least square method is proposed along with tricks to take advantage of the extremely sparse nature of the tensor. We can use this method in our computation of reward function, as discussed in the section 6

- *Shortcomings*: Limited type of queries can be performed.

HAR [5]

- *Main idea*: This paper provides a generalization of the classic PageRank algoirthm for multi-relational data. Basically, they view the multi-relational data as a labelled directed multi-graph, where edges are labelled with type of relation. Then they considered a random walker in the multi-graph to yield hub, authority and relevance scores (similar to TOPHITS) for nodes. These three scores are interleaved as high hub score vertex that points to many objects with high authority scores through relations of high relevance.

  The random walk on the multi-graph has three attributes when visiting any particular object: hub, authority and relevance, denote them by random variables X_t, Y_t and Z_t. Thus we need to have three transition probabilities $\Pr\{X_t = i_1 | Y_t = i_2, Z_t = j\}$, $\Pr\{Y_t = i_2 | Z_t = j, X_t = i_1\}$, $\Pr\{Z_t = j | Y_t = i_2, X_t = i_1\}$ which can be estimated as following tensors:

$$\begin{aligned}\mathfrak{H}(i_1,i_2,j) &= \frac{\mathfrak{X}(i_1,i_2,j)}{\sum_{i_1=1}^{n}\mathfrak{X}(i_1,i_2,j)}\\ \mathfrak{A}(i_1,i_2,j) &= \frac{\mathfrak{X}(i_1,i_2,j)}{\sum_{i_2=1}^{n}\mathfrak{X}(i_1,i_2,j)}\\ \mathfrak{R}(i_1,i_2,j) &= \frac{\mathfrak{X}(i_1,i_2,j)}{\sum_{j=1}^{m}\mathfrak{X}(i_1,i_2,j)}\end{aligned} \tag{8}$$

Like PageRank, again the HAR score will satisfy the following equations:

$$\begin{aligned}\bar{\mathbf{x}} &= (1-\alpha)\mathbf{o} + \alpha\mathfrak{H}\bar{\mathbf{y}}\bar{\mathbf{z}}\\ \bar{\mathbf{y}} &= (1-\beta)\mathbf{o} + \beta\mathfrak{A}\bar{\mathbf{x}}\bar{\mathbf{z}}\\ \bar{\mathbf{z}} &= (1-\gamma)\mathbf{r} + \gamma\mathfrak{R}\bar{\mathbf{x}}\bar{\mathbf{y}}\end{aligned} \tag{9}$$

where $\mathbf{o}$ and $\mathbf{r}$ are probability distributions over webpages and relations. The paper proves existence of solution and provides algorithm to solve this system of equations. These vectors contain the hub, authority and relevance scores for each vertex and relation respectively for the given query.

- *Use for our project*: Idea to represent triplet data as a multi-graph
- *Shortcomings*: Makes a questionable assumption about joint stationary distribution of $\mathbf{x}, \mathbf{y}, \mathbf{z}$ being independent.

  The so-called "interesting" objects are not defined/described properly, which are used the construct the $\mathbf{o}$ and $\mathbf{r}$ vectors. Moreover, this way we obtain the most relevant vertices to our query, and not the most interesting ones!

4 Dataset

In this section, we provide the background for our choice of datasets and toolboxes used in our experiments.

4.1 Datasets

We test our methods on two datasets of different nature. It is difficult to measure interestingness or surprise factor of a fact objectively. To best of our knowledge, we could not find any method for the task. We employ an ad hoc to measure: We fix a set of queries and run all the proposed method for that set. Then, we subjectively judge if we the facts returned by the algorithm on the fixed query set is surprising or not. We describe these two datasets below and list the fixed query set.

4.1.1 Nell KB

To test our method, we created a three-mode tensor using the publicly available NELL knowledge base. Each fact in the NELL knowledge base consists of a subject, a relation, an object, and a score between 0 and 1 indicating the confidence of the NELL learning system in the fact. The datasets are available at the NELL project website[1]. We construct a three-mode tensor corresponding to the subjects, relations, and objects. The universe from which subject and object entities are picked is the same. The entries in the tensor are real-valued between 0 and 1. The $(i, j, k)^th$ entry in the tensor corresponds to the triplet $subject_i - relation_j - object_k$ in the NELL KB. The the $(i, j, k)^th$ entry is populated with the confidence score of the fact $subject_i - relation_j - object_k$ in the NELL KB.

Table 2: Query Set

Subject/Object (Nouns)	**Relation (Verbs)**
graphical_models	attractionofcity
mahatma_gandhi	weaponmadeincountry
nelson_mandela	countryalsoknownas
pnc	wineryproduceswine
research	bankboughtbank

4.1.2 Yago KB

To test our method, we create a three-dimensional tensor using the publicly available Yago knowledge base. The knowledge base is available from MPI's website[2]. The dataset is provided in the form of subject-rel-object triplets, where *rel* is an asymmetric relation between subject and object. We construct a tensor which has three modes - one each for subjects,

[1] http://rtw.ml.cmu.edu/resources/results/08m/?C=S;O=D

[2] www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/

relations, and objects. The entries in the tensor are binary. If the $(i, j, k)^th$ entry in the tensor is 1, then the fact $subject_i - relation_j - object_k$ exists in the Yago KB. Similarly, 0 in the $(i, j, k)^th$ entry indicates that the fact $subject_i - relation_j - object_k$ was not mentioned in the KB.

Table 3: Query Set

Subject/Object (Nouns)	**Relation (Verbs)**
Barak Obama	influences
Carnegie Mellon University	hasAcademicAdvisor
Stanford University	actedIn
California	hasWonPrize
Napa County, California	holdsPoliticalPosition

4.1.3 Cleanup

After procuring the NELL and Yago datasets and extracting sparse tensors from these datasets, we cleaned up certain dominant and uninformative tuples like "hasWebsite" and "hasWikipediaUrl" which are one-to-one relations and not very helpful in discovering information structure. We refer to the cleaned tensors obtained as the NELL and Yago datasets as *NellTensor* and *YagoTensor* respectively.

4.2 Tools

We used Matlab to prototype our methods. To deal with the highly sparse nature of the Yago tensor, we use the Tensor Toolbox[3] provided by Dr. Kolda of Sandia National Laboratories. It provides an immensely useful implementation of various operators on sparse tensors. We note that our methods can be useful with dense tensors as well as long as the tensor factors are available using a toolbox for dense tensors, such as the N-Way Tensor Toolbox.

We also use ParCube, a system for decomposition of a sparse tensor into sparse factors. ParCube (which was surveyed in the related works section) makes it possible to parallelize the Kruskal decomposition of a sparse tensor by decomposing various subtensors of the original

[3] http://www.sandia.gov/ tgkolda/TensorToolbox/index-2.5.html

tensor and stitching the results together to give the Kruskal decomposition of the original tensor. While ParCube did offer significant advantages in terms of speedup, we observed that some of the values in the tensor factors obtained using this method were extremely huge or *NaN*. This was not the case with the `cp_als` method provided in the Tensor Toolbox. Thus, we found that the results from ParCube on the Yago and NELL tensor were not very suitable for explanatory mining of interesting aspects from the tensors.

5 Interestingness: CF-ICF

5.1 Description

We want an indication of the interestingness or the surprise of a subject-verb-object triplet. As discussed in the introduction, for interestingness we are looking for facts consisting of an obscure fact of popular entity. For this purpose, we generalize the popular concept of term frequencyinverse document frequency (tf-idf), which is a numerical statistic that is intended to reflect how important a word is to a document. According to our definition of interestingness among all the facts related to the query, we want to get the fact for which one of the term is very popular (occurs with a high frequency) and another terms occurs very rarely. So we can say that we want to choose the fact that maximizes the product of Corpus-Frequency(CF) and Inverse-Corpus-Frequency (ICF).

Moreover, we want to get rid of all global topics/concepts present in the knowledgebase. Because anything that can be inferred from a global structure would not be interesting. For this purpose we decompose the tensor, then each rank-1 factor would correspond to a global concept explaining the entries of the tensor. Once we subtract the contribution from such rank-1 factors, we are effectively removing the simple global low-rank structure from the tensor and capturing the more interesting local effects in the residual tensor. We define the residual tensor as the original tensor minus its rank-1 factors. Since we only care about the entries in the original tensor, we also project the residual tensor to be non-zero only at the entries where the original tensor was non-zero.

We can find out the low-rank tensor approximation by carrying out ALS/ParCube on the data tensor to obtain:

$$\mathfrak{X} \approx \sum_{r=0}^{R} \lambda_r \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \tag{10}$$

To calculate the reward function, we would like to find localized dense clusters of subject-verb-object interactions which are not explained by the low-rank tensors that approximate the entire tensor. Such clusters will emerge when we subtract the tensor decomposition factors from the original tensor to get the residual tensor. Our analysis reveals that such a

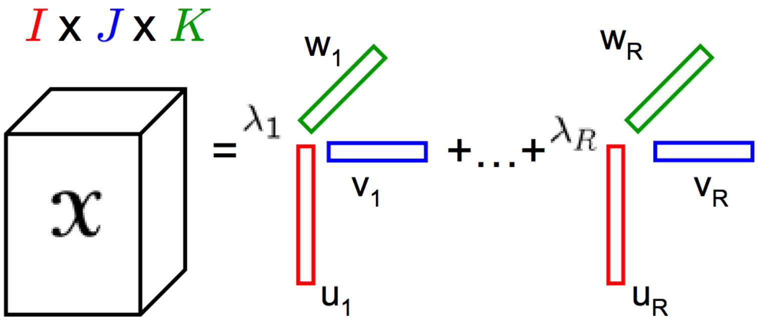


Figure 2: Kruskal Decomposition of Tensor $\mathfrak{X}$

residual tensor will not be sparse and hence we do not want to instantiate the residual tensor. Instead we only calculate the residual tensor at the indices where the original tensor had non-zero values. This is natural since we only want to calculate the interestingness (through the residual) of subject-verb-object triplets that actually appeared in the dataset. If A is the set of indices where the original tensor had non-zero entries, we calculate the residual as follows:

$$\mathfrak{R} = \Pi_A \left(\mathfrak{X} - \sum_{r=0}^{R} \lambda_r \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \right)$$

where the term inside the brackets is the actual residual and $\Pi_A(\cdot)$ is the projection of the actual residual on the non-zero entries of the original tensor. Once we have the sparse residual, we expect to see localized clusters of subject-verb-object triplets in the residual since the low-rank structure common to the entire tensor has been subtracted out. We have thought of efficient ways to mine these clusters on massive tensors and use them to calculate the reward function for the MDP iteration.

We consider reward functions based on *specificity.* To describe the reward function, we need to specify the value of function $r(s, a, s')$ for each source state s, destination state s', and action a' which takes us from s to s'. In the case of the Yago tensor, the universe of states corresponds to the entities in the Yago KB and the universe of actions corresponds to the possible relations between two entities. Similarly, in the NELL dataset, the states correspond to the subjects and objects in the tuples while the actions correspond to the verbs in the tuples between the subject and object. We consider the following reward function or interestingness score:

$$r(s,a,s') = \min\left\{\mathfrak{X}(s,a,s') \cdot \frac{\sum_{\forall s'} \mathfrak{X}(s,a,s')}{\sum_{\forall a} \mathfrak{X}(s,a,s')}, \mathfrak{X}(s,a,s') \cdot \frac{\sum_{\forall a} \mathfrak{X}(s,a,s')}{\sum_{\forall s'} \mathfrak{X}(s,a,s')}\right\} \tag{11}$$

To understand this score, consider $\mathfrak{X}(s,a,s') \cdot \frac{\sum_{\forall s'} \mathfrak{X}(s,a,s')}{\sum_{\forall a} \mathfrak{X}(s,a,s')}$. By minimizing this expression for a given s, we are finding a and s' that jointly minimizes $\mathfrak{X}(s,a,s')$, minimizes $\sum_{\forall s'} \mathfrak{X}(s,a,s')$ and maximizes $\sum_{\forall a} \mathfrak{X}(s,a,s')$. This corresponds to finding the a and s' pair that is the most specific to s, while maximizing the specificity of a and minimizing the specificity of s'. We have chosen this interestingness function because we want to discover a relatively unknown aspect of a well-known subject as this will be fairly surprising to most human evaluators. Spotting obscure facts about an obscure entity is not as interesting to humans as spotting an obscure fact about a popular entity. The explanation for the second similar part of the interestingness function $\mathfrak{X}(s,a,s') \cdot \frac{\sum_{\forall a} \mathfrak{X}(s,a,s')}{\sum_{\forall s'} \mathfrak{X}(s,a,s')}$ follows a similar explanation where we are looking for minimizing the specificity of a and maximizing the specificity of s'.

Due to numerical instability in the factors obtained from ParCube, we chose to use the tensor factors provided by the `cp_als` routine in the further steps of our experiments. `cp_als` of *YagoTensor* gives a *ktensor* object in Matlab, which stores a tensor as a Kruskal decompsition and provides any entry of the tensor by computing it on the fly from the factors. Kruskal decomposition of a tensor represents a tensor as a sum of rank-1 tensors, which are obtained from the outer product of a vector along each mode. It is illustrated in figure 2.

5.2 Experimental Results

5.2.1 Yago - Original Tensor

We calculate the projected residual of the tensor $\mathcal{T}$ from its `cp_als` decomposition $\mathcal{P}$ as follows:

$$\Pi(\mathcal{P}) = \mathcal{T} - \mathcal{T}. * \mathcal{P} \tag{12}$$

where $.*$ indicates the elementwise product between tensors of identical dimensions. Since $\mathcal{T}$ and $\mathcal{P}$ are both stored as special sparse tensors, the elementwise dot-product can be performed without leading to an out-of-memory error. $\mathcal{T}. * \mathcal{P}$ is the projection of the tensor reconstruction $\mathcal{P}$ onto the non-zero entries of the original tensor $\mathcal{T}$. Finally, we subtract this projection of the tensor reconstruction from the original tensor $\mathcal{T}$ to get the projected residual $\Pi(\mathcal{P})$.

Here, we use the *YagoTensor* dataset. In the interestingness function, we use the *YagoTensor* $\mathcal{T}$ in place of $\mathfrak{X}$. The results are presented in Table 4.

5.2.2 Yago - Projected Residual

Here, we again use the *YagoTensor* dataset. In the interestingness function, we use $-\Pi(\mathcal{P})$ in place of $\mathfrak{X}$ where $\Pi(\mathcal{P})$ is the projection of the residual of the *YagoTensor* on the non-zero entries in the *YagoTensor*. We found that using the projected residual gives more interesting/surprising results on our evaluation set. The results are presented in Table 5.

5.2.3 NELL - Original Tensor

Here, we use the *NellTensor*. In the interestingness function, we use the *NellTensor* $\mathcal{T}$ in place of $\mathfrak{X}$. We find that our method detects the most surprising facts from the NELL dataset – facts that are in the NELL Knowledge Base but we have verified to be actually false. It makes sense that these false facts are returned as interesting/surprising results from the *NellTensor*. We believe that our method can be used to point out NELL facts whose validity and confidence needs to be verified by a human evaluator. The results are presented in Table 6.

5.2.4 NELL - Projected Residual

The results using the projected residual of the *NellTensor* are reported in table 7. In the interestingness function, we use $-\Pi(\mathcal{P})$ in place of $\mathfrak{X}$ where $\Pi(\mathcal{P})$ is the projection of the residual of the *NellTensor* on the non-zero entries in the *NellTensor*.

5.3 Analysis of Results

We find that using specificity of a fact as a measure of interestingness yields some good results that are interesting to the person who wants to know about the entities involved in those facts. We also note that sometimes tensor decomposition is not the best way to find out latent structure. Consider the matrix slice obtained from the Yago tensor by restricting to the “hasWebsite” relation. The sparsity pattern of this matrix slice is visualized in figure 4. We see that the two dimensions are not independent at all. Hence, a low-rank decomposition will not be able to reconstruct such a matrix slice or the tensor that contains it. Further analysis on the matrix slice also revealed that it also doesn’t obey any obvious power laws in terms of the distribution of frequencies of magnitudes.

Also, due to the dominance of sparse random noise in the tensor, low-rank decomposition using 100 factors led to poor reconstruction accuracy. The ratio of the norm of the projected residual and the original tensor was $\frac{4.9}{5.1}$, indicating that much of the noise of the original tensor was still present in the residual. This points to the fact that the original tensor has a lot of sparse noise to begin with. A manual inspection of the facts in the NELL KB confirms this.

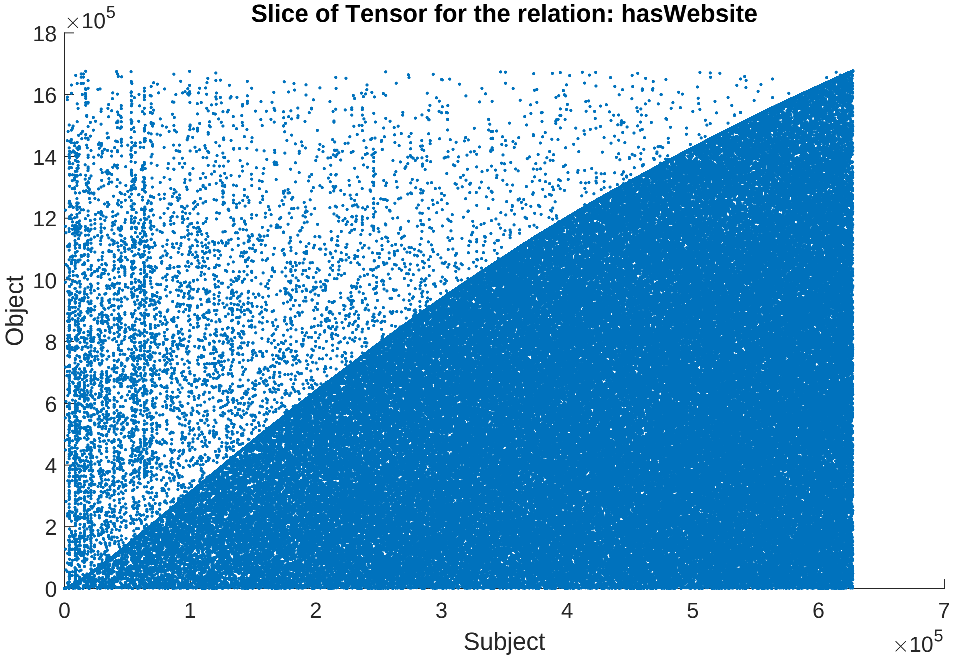


Figure 3: Matrix Slice for "hasWebsite" relation

6 Importance: MDP

6.1 Description

This approach is motivated from the success of PageRank algorithm. The PageRank provides an importance score for all vertices in a normal graph. Unfortunately, the PageRank algorithm only works for matrices and its extension for tensors is non-trivial. But, if we observe a big similarity between the Bellman equation for MDP and the PageRank equation and thus look at Bellman equation as a generalization of PageRank for multi-graph with score given by the value vector rather than the eigenvector. Next, we describe this connection in more details.

Table 4: Examples of interesting facts from Yago KB using original Tensor

	Query	Outcome
Subject	Barack Obama	Barack Obama created Of Thee I Sing (book) Barack Obama created The Audacity of Hope Barack Obama created We Are the Ones
	Carnegie Mellon University	Carnegie Mellon University created GraphLab Carnegie Mellon University created CMU Pronouncing Dictionary Carnegie Mellon University created Mach (kernel)
	Stanford University	Stanford University created Storage@home Stanford University created Stanford Shopping Center Stanford University created EteRNA
	California	California hasWebsite http://www.ca.gov/ California isLocatedIn United States California participatedIn Pitt River Expedition
	Napa County, California	Napa County California isLocatedIn United States Napa County California isLocatedIn San Francisco Bay Area Napa County California isLocatedIn California
Relation	influences	Elvis Presley influences Samuel Hui The Beatles influences Donald Gallinger The Beatles influences Gerwin van der Werf
	hasAcademicAdvisor	Gottfried Leibniz hasAcademicAdvisor Erhard Weigel Andrew Hill hasAcademicAdvisor Fulvio Melia Richard Dawkins hasAcademicAdvisor Nikolaas Tinbergen
	actedIn	David Bowie actedIn Everybody Loves Sunshine Roscoe Arbuckle actedIn Killing Horace Roscoe Arbuckle actedIn The Baggage Smasher
	hasWonPrize	Gnther Mller-Stckheim hasWonPrize Knight's Cross of the Iron Cross Hans Bartels hasWonPrize Knight's Cross of the Iron Cross Richard Ruoff hasWonPrize Knight's Cross of the Iron Cross
	holdsPoliticalPosition	Alfred Boultbee holdsPoliticalPosition Member of Provincial Parliament (Ontario) Oliver Aiken Howland holdsPoliticalPosition Member of Provincial Parliament (Ontario) Thomas Fraser (Upper Canada politician) holdsPoliticalPosition Member of Provincial Parliament (Ontario)
Object	Stanford University	Sara Hall isAffiliatedTo Stanford University Foluke Akinradewo isAffiliatedTo Stanford University Alix Klineman isAffiliatedTo Stanford University
	Carnegie Mellon University	Demetri Psaltis graduatedFrom Carnegie Mellon University Peter A. Freeman graduatedFrom Carnegie Mellon University James Knepper graduatedFrom Carnegie Mellon University
	Barack Obama	Barack Obama Sr. hasChild Barack Obama Ann Dunham hasChild Barack Obama Ann Dunham hasChild Barack Obama
	California	Powellton Meadow California isLocatedIn California Bradtmoore California isLocatedIn California Tsuka California isLocatedIn California
	Napa County, California	Lake Berryessa isLocatedIn Napa County California Beaulieu Vineyard isLocatedIn Napa County California Napa County Airport isLocatedIn Napa County California

Table 5: Examples of interesting facts from Yago KB using residual Tensor

	Query	Outcome
Subject	Barack Obama	Barack Obama created Of Thee I Sing (book) Barack Obama created The Audacity of Hope Barack Obama created We Are the Ones
	Carnegie Mellon University	Carnegie Mellon University created Emergent (software) Carnegie Mellon University created EteRNA Carnegie Mellon University created GraphLab
	Stanford University	Stanford University isLocatedIn Stanford California Stanford University created Storage@home Stanford University created Self (programming language)
	California	California hasWebsite http://www.ca.gov/ California owns Transportation in Visalia California owns Keynote Sigos
	Napa County, California	Napa County California isLocatedIn United States Napa County California isLocatedIn California Napa County California isLocatedIn United States
Relation	influences	Elvis Presley influences Samuel Hui The Beatles influences Donald Gallinger The Beatles influences Gerwin van der Werf
	hasAcademicAdvisor	Gottfried Leibniz hasAcademicAdvisor Erhard Weigel Andrew Hill hasAcademicAdvisor Fulvio Melia Richard Dawkins hasAcademicAdvisor Nikolaas Tinbergen
	actedIn	Mizuo Peck actedIn Night at the Museum Neva Small actedIn Fiddler on the Roof (film) Jennifer Elise Cox actedIn The Brady Bunch Movie
	hasWonPrize	Bentley Kassal hasWonPrize Bronze Star Medal Charles S. Schepke hasWonPrize Medal of Honor Alice Pearce hasWonPrize Emmy Award
	holdsPoliticalPosition	Louis H. Pollak holdsPoliticalPosition United States District Court for the Eastern District of Pennsylvania Irving Lehman holdsPoliticalPosition Chief judge Alfred Boultbee holdsPoliticalPosition Member of Provincial Parliament (Ontario)
Object	Stanford University	Robert S. Smith graduatedFrom Stanford University Gene D. Block graduatedFrom Stanford University Gerard Davis isAffiliatedTo Stanford University
	Carnegie Mellon University	Demetri Psaltis graduatedFrom Carnegie Mellon University Peter A. Freeman graduatedFrom Carnegie Mellon University Ravi Jagannathan graduatedFrom Carnegie Mellon University
	Barack Obama	Ann Dunham hasChild Barack Obama Barack Obama Sr. hasChild Barack Obama Ann Dunham hasChild Barack Obama
	California	Venice Los Angeles isLocatedIn California Echo Park Los Angeles isLocatedIn California Rosamond Lake isLocatedIn California
	Napa County, California	Lake Berryessa isLocatedIn Napa County California Beaulieu Vineyard isLocatedIn Napa County California Napa County Airport isLocatedIn Napa County California

Table 6: Examples of interesting facts from NELL KB using original Tensor

	Query	Outcome
Subject	graphical_models	graphical_models mlareaexpert nir_friedman graphical_models mlareaexpert david_heckerman graphical_models generalizations mlalgorithm
	mahatma_gandhi	mahatma_gandhi organizationacronymhasname mohandas_karamchand_gandhi mahatma_gandhi actorstarredinmovie ali_jinnah mahatma_gandhi personborninlocation rajghat
	nelson_mandela	nelson_mandela agentcreated robben_island nelson_mandela politicianusendorsedbypoliticianus oliver_tambo nelson_mandela atdate n1999
	pnc	pnc:acquired:washington_st_louis pnc parkincity holmdel pnc bankchiefexecutiveceo larry_zimpleman
	research	research sportusesequipment research_question research sportusesequipment scientific_developments research agentcompeteswithagent revolvers
Relation	attractionofcity	london attractionofcity london_institute london attractionofcity n32_london london attractionofcity london_institute
	weaponmadeincountry	icbm_force weaponmadeincountry united_states north_korean_missile weaponmadeincountry united_states n1973_oil weaponmadeincountry united_states
	countryalsoknownas	london countryalsoknownas rhodesian australia countryalsoknownas hutt_river_province england countryalsoknownas hun
	wineryproduceswine	lemon wineryproduceswine roasted_shallots lemon_plant wineryproduceswine roasted_shallots handcraft wineryproduceswine three
	bankboughtbank	department bankboughtbank economic_security uk bankboughtbank uk_debt microsoft bankboughtbank musiwave
Object	graphical_models	bayes sportusesequipment graphical_models - -
	mahatma_gandhi	kottayam cityliesonriver mahatma_gandhi ali_jinnah hashusband mahatma_gandhi hindu_extremist hasbrother mahatma_gandhi
	nelson_mandela	a_long_walk_to_freedom bookwriter nelson_mandela winnie_mandela hashusband nelson_mandela bishop_desmond_tutu politicianusendorsedbypoliticianus nelson_mandela
	pnc	james_e_rohr ceoof pnc pnc_bank teamhomestadium pnc pnc_bank teamhomestadium pnc
	research	eclecticism agentcompeteswithagent research human_diets sportusesequipment research academic_events sportusesequipment research

Table 7: Examples of interesting facts from NELL KB using residual Tensor

	Query	Outcome
Subject	graphical_models	graphical_models mlareaexpert nir_friedman graphical_models mlareaexpert david_heckerman graphical_models generalizations mlalgorithm
	mahatma_gandhi	mahatma_gandhi organizationacronymhasname mohandas_karamchand_gandhi mahatma_gandhi actorstarredinmovie ali_jinnah mahatma_gandhi personborninlocation rajghat
	nelson_mandela	nelson_mandela politicianusendorsedbypoliticianus oliver_tambo nelson_mandela atdate n1999 nelson_mandela agentcreated robben_island
	pnc	pnc:acquired:washington_st_louis pnc parkincity holmdel pnc bankchiefexecutiveceo larry_zimpleman
	research	research sportusesequipment research_question research sportusesequipment scientific_developments research agentcompeteswithagent revolvers
Relation	attractionofcity	soho_square attractionofcity london london attractionofcity london_institute barbican_hall attractionofcity london
	weaponmadeincountry	north_korean_missile weaponmadeincountry united_states gay_marriages weaponmadeincountry california n1973_oil weaponmadeincountry united_states
	countryalsoknownas	london countryalsoknownas rhodesian australia countryalsoknownas hutt_river_province england countryalsoknownas hun
	wineryproduceswine	lemon wineryproduceswine roasted_shallots blue wineryproduceswine mountain_chardonnay handcraft wineryproduceswine three
	bankboughtbank	department bankboughtbank economic_security uk bankboughtbank uk_debt microsoft bankboughtbank musiwave
Object	graphical_models	bayes sportusesequipment graphical_models - -
	mahatma_gandhi	kottayam cityliesonriver mahatma_gandhi ali_jinnah hashusband mahatma_gandhi obama agentcollaborateswithagent mahatma_gandhi
	nelson_mandela	a_long_walk_to_freedom bookwriter nelson_mandela winnie_mandela hashusband nelson_mandela bishop_desmond_tutu politicianusendorsedbypoliticianus nelson_mandela
	pnc	james_e_rohr ceoof pnc education academicprogramatuniversity pnc pnc_bank teamhomestadium pnc
	research	eclecticism agentcompeteswithagent research human_diets sportusesequipment research academic_events sportusesequipment research

As, discussed in the Section 3, in calculating PageRank score of a node the assumption is that edges to a vertex indicates its importance. Therefore a vertex with many incoming links has high importance. So a simple measure of importance is a count of a pages incoming edges (its indegree). But a more sensible task would be weight the importance of edge by the importance of node from which the egde is originating. So an edge from a vertex that has large PageRank, contributes more than a link from a page with low PageRank (all other things being equal). The effectiveness of PageRank algorithm has been time tested. Thus, one can conclude that a centrality measure (like PageRank) is a better score than simple measure of indegree count.

We hope to carry over the similar key idea to tensors/multi-graphs. As discussed in Section 5 that TF-CF-ICF provides a simplistic measure and using that we want to find a centrality measure. The MDP with its value vector appeared to be a good candidate for this purpose.

For this, we consider the process random walk on this multi-graph to find most interesting triplet. Suppose we are at a vertex s, then we can select a type of edge (basically verb) and then randomly pick an edge from that type to proceed to a new vertex. With this heuristic, we can employ MDP to come up with a solution.

As discussed in literature survey, the optimal policy of MDP is obtained through the value iteration algorithm which tries to find a fixed point solution to the following equation:

$$\mathbf{v}^* = \max_{a \in \mathcal{A}} \left[r(:, a) + \gamma \mathcal{T}(:, a, :) \mathbf{v}^* \right] \tag{13}$$

This can be thought of as a generalization of the PageRank algorithm which also tries to find a fixed point solution to the following equation:

$$\mathbf{r}^* = \frac{(1-d)}{n}\mathbf{1} + d\mathbf{W}\mathbf{r}^* \tag{14}$$

with reward being constant. Note the similarity between the two equations.

In particular,

$$\mathbf{v}^*(s) = \max_{a \in \mathcal{A}} \left[r(s, a) + \gamma \mathcal{T}(s, a, :) \cdot \mathbf{v}^* \right] \tag{15}$$

where $\mathbf{v}^*(s)$ is the value of being in state s. $r(s, a)$ indicates the reward that we are granted for taking action a from a certain state s. $\mathcal{T}(s, a, :)$ is a vector which encodes the transition probabilities from state s on action a to any other action indicated by the placeholder :. $\mathbf{v}^*$ indicates the vector of values of the states we have transitioned into. γ indicates the decay factor of the earned reward with time.

In the process of value iteration, we will not need to perform more than $O(l)$, where l is the number of entries in the data tensor $\mathfrak{X}$. This can be more clearly seen if we rewrite the recursion as:

$$\mathbf{v}^*(s) = \max_{a \in \mathcal{A}} \sum_{s^+ \in \mathcal{S}} \left[\mathcal{R}(s, a, s^+) \tau(s, a, s^+) + \gamma \mathbf{v}^*(s^+) \tau(s, a, s^+) \right] \tag{16}$$

The element-wise multiplication with $\mathcal{T}$ ensures that only terms corresponding to non-zero $\mathcal{T}$ play a role.

6.2 Experimental Results

We ran the MDP value iteration as shown in Algorithm 1 for both Yago KB and Nell KB using the reward function defined in previous section. For Yago, we found that applying the reward function and choosing the verb-object pair that minimizes $r(subject, verb, object)$ for a given subject gives fairly important verb-object pairs for that particular subject. Finding the most important subject-verb pair for a given object and the most important subject-object pair for a given verb can be done in a similar fashion. Our results for all these scenarios are presented in the Table 8. Also for NELL, we performed similar experiments and the result is shown in Table 10.

Algorithm 1 Infinite Horizon Value Iteration

Input: The Markov model Ξ

Output: Value function $\mathbf{v}^*$ and Optimal policy π^*

```
Initialize $\mathbf{v}^*(s) \leftarrow \infty$ for all $s \in \mathcal{S}$.
Initialize $\mathbf{v}^*_{\text{new}}(s) \leftarrow 0$ for all $s \in \mathcal{S}$.
while $\|\mathbf{v}^* - \mathbf{v}^*_{\text{new}}\| < \epsilon$ do
    $\mathbf{v}^* \leftarrow \mathbf{v}^*_{\text{new}}$
    for $s \in \mathcal{S}$ do
        for $a \in \mathcal{A}$ do
            $Q(s, a) = \sum_{s^+ \in \mathcal{S}} \left[ \mathcal{R}(s, a, s^+) \tau(s, a, s^+) + \gamma \mathbf{v}^*(s^+) \tau(s, a, s^+) \right]$
        end for
        $\mathbf{v}^*_{\text{new}}(s) = \max_{a \in \mathcal{A}} Q(s, a)$
        $\pi^*(s) = \arg\max_{a \in \mathcal{A}} Q(s, a)$
    end for
end while
```

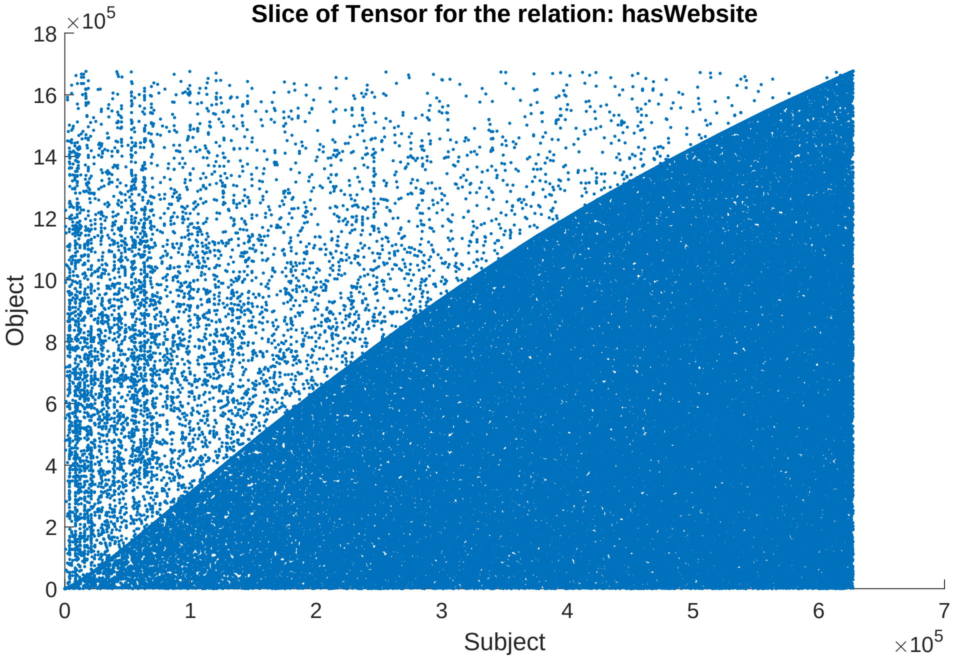


Figure 4: Matrix Slice for "hasWebsite" relation

6.3 Analysis

By visual inspection, the result yielded by the proposed MDP do look the important terms related to the query term. The result from Nell does not look impressive, but upon manually exploring those entries we found out there was not any other fact present which could be have been deemed more important.

We compare our results with the HAR model on the Yago KB query set only. The HAR is implemented as in Algorithm 2. The results of HAR model are given in Table 9. Comparing the two, it looks like facts returned by MDP seem to be much more important than the those returned by HAR.

7 Coherence: BTB

In this section, we describe the *Bayesian Tensor Blockmodel (BTB)*, a method that can discover and characterize dense and semantically coherent local blocks in the tensor. We

Table 8: Examples of important facts from Yago KB using MDP

	Query	Outcome
Subject	Barack Obama	Barack Obama isLeaderOf United States
	Carnegie Mellon University	Carnegie Mellon University isLocatedIn United States
	Stanford University	Stanford University isLocatedIn Stanford California
	California	California isLocatedIn United States
	Napa County, California	Napa County California isLocatedIn United States
Relation	influences	H.P. Lovecraft influences Rod Serling
	hasAcademicAdvisor	Richard Feynman hasAcademicAdvisor John Archibald Wheeler
	actedIn	Harold Lloyd actedIn Girl Shy
	hasWonPrize	Rod Serling hasWonPrize Golden Globe Award
	holdsPoliticalPosition	Arnold Schwarzenegger holdsPoliticalPosition Governor of California
Object	Barack Obama	Ann Dunham hasChild Barack Obama
	Carnegie Mellon University	Richard Duffin worksAt Carnegie Mellon University
	Stanford University	Donald Knuth worksAt Stanford University
	California	Battle of Olompali happenedIn California
	Napa County, California	Lake Berryessa isLocatedIn Napa County California

Algorithm 2 The HAR Algorithm for subject query

Input: The HAR model: tensors $\mathfrak{H}, \mathfrak{A}$ and $\mathfrak{R}$ and query vector $\mathbf{s}$

Output: Three limiting probability distributions $\mathbf{x}$(hub scores), $\mathbf{y}$ (authority scores) and $\mathbf{z}$ (relevance values)

```
Initialize $\mathbf{x}(s), \mathbf{y}(s) \leftarrow \infty$ for all $s \in \mathcal{S}$.
Initialize $\mathbf{z}(a) \leftarrow \infty$ for all $a \in \mathcal{A}$.
Initialize $\mathbf{x}_{\text{new}}(s), \mathbf{y}_{\text{new}}(s) \leftarrow 0$ for all $s \in \mathcal{S}$.
Initialize $\mathbf{z}_{\text{new}}(a) \leftarrow 0$ for all $a \in \mathcal{A}$.
Set $\mathbf{r} \leftarrow \mathbf{1}/m, \quad \mathbf{o} \leftarrow \mathbf{1}/n$.
while $\|\bar{\mathbf{x}} - \bar{\mathbf{x}}_{\text{new}}\| + |\bar{\mathbf{y}} - \bar{\mathbf{y}}_{\text{new}}\| + |\bar{\mathbf{z}} - \bar{\mathbf{z}}_{\text{new}}\| < \epsilon$ do
    $\bar{\mathbf{x}} \leftarrow \bar{\mathbf{x}}_{\text{new}}, \quad \bar{\mathbf{y}} \leftarrow \bar{\mathbf{y}}_{\text{new}}, \quad \bar{\mathbf{z}} \leftarrow \bar{\mathbf{z}}_{\text{new}}$
    $\bar{\mathbf{x}}_{\text{new}} \leftarrow (1-\alpha)\mathbf{s} + \alpha\mathfrak{H}\bar{\mathbf{y}}\bar{\mathbf{z}}$
    $\bar{\mathbf{y}}_{\text{new}} \leftarrow (1-\beta)\mathbf{o} + \beta\mathfrak{A}\bar{\mathbf{x}}_{\text{new}}\bar{\mathbf{z}}$
    $\bar{\mathbf{z}}_{\text{new}} \leftarrow (1-\gamma)\mathbf{r} + \gamma\mathfrak{R}\bar{\mathbf{x}}_{\text{new}}\bar{\mathbf{y}}_{\text{new}}$
end while
```

Table 9: Examples of important facts from Yago KB using HAR

	Query	Outcome
Subject	Barack Obama	Barack Obama isMarriedTo Michelle Obama
	Carnegie Mellon University	Carnegie Mellon University owns Death Risk Rankings
	Stanford University	Stanford University created SRI International
	California	California hasCapital Sacramento California
	Napa County, California	Napa County California isLocatedIn San Francisco Bay Area
Relation	influences	Karl Marx influences Bill Mitchell (economist)
	hasAcademicAdvisor	Benjamin C. Pierce hasAcademicAdvisor Robert Harper (computer scientist)
	actedIn	Harold Lloyd actedIn Speedy (film)
	hasWonPrize	Anthony Davis (basketball) hasWonPrize NABC Defensive Player of the Year
	holdsPoliticalPosition	Lothar de Maizire holdsPoliticalPosition Council of Ministers of the GDR
Object	Barack Obama	Michelle Obama isMarriedTo Barack Obama
	Carnegie Mellon University	David Brumley worksAt Carnegie Mellon University
	Stanford University	Logan Tom playsFor Stanford University
	California	Klamath Falls Airport isConnectedTo California
	Napa County, California	Lisa Gansky livesIn Napa County California

Table 10: Examples of interesting facts from NELL KB using original Tensor

	Query	Outcome
Subject	graphical_models	graphical_models mlareaexpert daphne_koller
	mahatma_gandhi	mahatma_gandhi persondeathdate n1947
	nelson_mandela	nelson_mandela personhascitizenship south_africa
	pnc	pnc generalizations bank
	research	research languageofcity washington_d_c
Relation	attractionofcity	thameslink attractionofcity london
	weaponmadeincountry	weapons_program weaponmadeincountry united_states
	countryalsoknownas	book:america countryalsoknownas aquarium:us
	wineryproduceswine	mission wineryproduceswine merlot
	bankboughtbank	kennedy bankboughtbank stewart

primarily use the NELL dataset to demonstrate the method since it is much larger and noisier than the Yago dataset.

7.1 Description

7.1.1 Tensor Decomposition

The first stage of BTB consists of finding a kruskal decomposition of the tensor. We tried two methods for finding the kruskal decomposition - the `cp_als` method provided in the Tensor Toolbox provided by Sandia National Laboratories, and the *ParCube* method. We noticed that the factors provided by ParCube show signs of numerical instability of the system. Particularly, we found that some of the entries in the factors provided by ParCube were either extremely large or NaN. Consequently, we proceeded with the results provided by the `cp_als` method. We extracted 100 rank-1 factors of the NELL tensor using `cp_als`. We used a server with 64GB RAM since decomposing the massive NELL tensor is not possible on a commodity workstation.

7.1.2 Data Generating Process for Bayesian Tensor Blockmodel

After obtaining the rank-1 factors, we perform a *Bayesian Tensor Blockmodel* analysis on each factor. We found that each factor contained many semantically distinct concepts and we believe that the BTB method will help detect and characterize these concepts.

To disentangle these hidden concepts within each factor, we employ a hierarchical bayesian model which described the generative mechanism of the factor. The model is shown in figure 5. The notation used in the figure is listed in table 11.

The data generating mechanism for a tensor block is as follows:

1. For each mode $m \in [1, .., M]$, we first sample a dirichlet parameter γ_{mb} (for a multinomial distribution over mode's indices) for each blockmodel $b \in [1, .., B]$ using the hyperparameter η_m.

2. For each mode $m \in [1, .., M]$, we also sample a dirichlet parameter ϕ_{mb} (for a multinomial distribution over words) for each blockmodel $b \in [1, .., B]$ using the hyperparameter α_m.

3. To generate a tensor entry, we first sample the block to which it belongs. We first sample a dirichlet parameter θ (for a multinomial distribution over blocks) using the hyperparameter β. Using multinomial parameter θ, to generate tensor entry $T_d,\ d \in [1, .., D]$, we first sample its block $z_d,\ d \in [1, .., D]$.

4. Once we know the block, we sample its indices along each mode using that particular block's distribution over indices γ_{mz_d}.

5. Further, we also sample the words describing the tensor entry along each mode using that particular block's distribution over words ϕ_{mz_d}. Unlike the case of indices where we sample only one index per mode, we can sample N words to describe the entry per mode as shown in the plate diagram.

We wrote code to perform posterior inference for the above bayesian tensor blockmodel. We obtained promising results which are presented in a later section.

Symbol	Definition
D	number of tensor entries
M	number of modes of the tensor
B	number of tensor blocks
N	number of words describing each dimension across each mode
β	hyperparameter for distribution over topics
θ	distribution over topics
z	topic assignment for a particular tensor entry
η	hyperparameter to generate distribution over indices
γ	distribution over mode's indices, one for each tensor block
I	Index across a mode sampled to generate tensor entry
α	hyperparameter to generate topics for each block across each mode
ϕ	topic used to explain a block along a mode
w	words sampled to explain

Table 11: Symbols and definitions

7.2 Experimental Results

We will present results of the Bayesian Tensor Blockmodel for one of the thresholded rank-1 factors (which we call T_f) of the NELL tensor later. For now, we note that it is possible to perform BTB analysis on T_f because it is a much smaller tensor and has around 3000 indices across each mode. We believe that this kind of analysis is suitable for massive tensors, where we first perform a simplistic tensor decomposition as a first step to find factors that capture much of the magnitude of the original tensor and then detect semantically coherent tensor blocks from each factor.

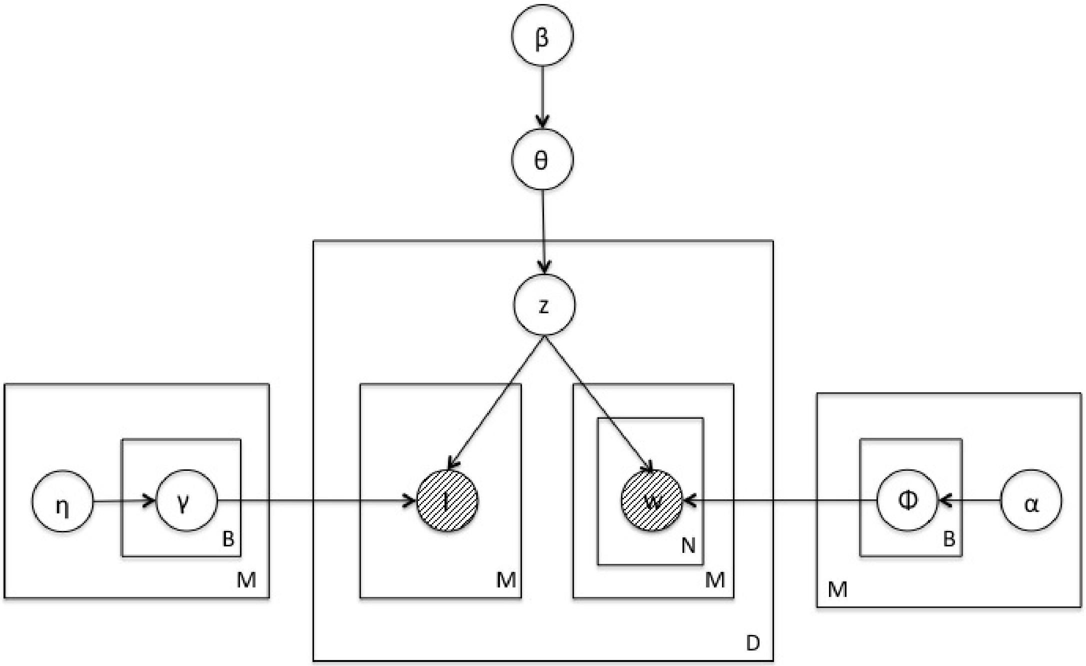


Figure 5: Bayesian Tensor Blockmodel Plate Diagram

We chose a truncated rank-1 factor for the NELL tensor and performed Bayesian Tensor Blockmodel analysis. We chose to detect 3 tensor blocks from the truncated factor. The unnormalized distribution over the three blocks is as follows: (5637, 4830, 13283).

7.3 Analysis

TBD

8 Conclusions and Future Work

We considered the challenging question of finding interesting local structure in tensors. We investigated the CF – ICF method for finding interesting triplets from a tensor using the data tensor as well as its projected residual. The method showed promising results as shown in the experiments. We also investigates the use of MDP as a generalization of PageRank to detect important triplets from a tensor, and found that the results were better compared to HAR, current state of the art.

We also described the Bayesian Tensor Blockmodel to detect dense semantically coherent tensor blocks in a massive tensor. This is still work in progress, and evaluation results using the method will be shared soon. We plan to improve the tensor decomposition component of BTB by replacing the Alternating Least Squares (ALS) method from Tensor Toolbox with a Distributed Stochastic Gradient Descent (DSGD) version that often provides better reconstruction accuracy with respect to the original tensor. We also plan to tweak the hyperparameters of the Bayesian Tensor Blockmodel to improve the detection of semantically coherent hidden tensor blocks.

References

[1] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1):107–117, 1998.

[2] U Kang, Evangelos Papalexakis, Abhay Harpale, and Christos Faloutsos. Gigatensor: scaling tensor analysis up by 100 times-algorithms and discoveries. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 316–324. ACM, 2012.

[3] Tamara G Kolda, Brett W Bader, and Joseph P Kenny. Higher-order web link analysis using multilinear algebra. In *Data Mining, Fifth IEEE International Conference on*, pages 8–pp. IEEE, 2005.

[4] Joonseok Lee, Seungyeon Kim, Guy Lebanon, and Yoram Singer. Local low-rank matrix approximation. In *Proceedings of The 30th International Conference on Machine Learning*, pages 82–90, 2013.

[5] Xutao Li, Michael K Ng, and Yunming Ye. Har: Hub, authority and relevance scores in multi-relational data for query search. SDM 2012.

[6] Sridhar Mahadevan and Mauro Maggioni. Proto-value functions: A laplacian framework for learning representation and control in markov decision processes. *Journal of Machine Learning Research*, 8(2169-2231):16, 2007.

[7] Evangelos E Papalexakis, Christos Faloutsos, and Nicholas D Sidiropoulos. Parcube: Sparse parallelizable tensor decompositions. In *Machine Learning and Knowledge Discovery in Databases*, pages 521–536. Springer, 2012.

A Appendix

A.1 Labor Division

The team performed the following tasks:

- Coming up with CF-ICF [Manzil]
- Formalizing random walks on Tensors [Manzil]
- Formalizing reward function describing “goodness” of triplets [Manzil]
- Implement scalable MDP Value iteration [Manzil]
- Implement Bayesian Tensor Blockmodel [Abinav]
- Explore Yago and Nell datasets and pre-processing [Abhinav and Manzil]
- Experiments on the real data [Abhinav and Manzil]
- Report writing and Software Packaging [Manzil]

A.2 Full disclosure with respect to dissertations/projects

Abhinav: He is not doing any project or dissertation related to this project. His first paper consists of detecting subtle emerging topics in real-world data streams.

Manzil: He is not doing any project or dissertation related to this project. His thesis is on Bayesian Non-parametrics and its relations to point process.

Contents