Exploration of Social and Web Image Search Results Using Tensor Decomposition

Liuqing Yang
University of California Riverside
Computer Science & Engineering
lyang030@ucr.edu

Evangelos E. Papalexakis
University of California Riverside
Computer Science & Engineering
epapalex@cs.ucr.edu

Abstract

How do socially popular images differ from authoritative images indexed by web search engines? Empirically, social images on e.g., Twitter often tend to look more diverse and ultimately more "personal", contrary to images that are returned by web image search, some of which are so-called "stock" images. Are there image features, that we can automatically learn, which differentiate the two types of image search results, or features that the two have in common? This paper outlines the vision towards achieving this result. We propose a tensor-based approach that learns key features of social and web image search results, and provides a comprehensive framework for analyzing and understanding the similarities and differences between the two types types of content. We demonstrate our preliminary results on a small-scale study, and conclude with future research directions for this exciting and novel application.

1. Introduction

Given a search query of an object or scene of interest, such as "Golden Gate Bridge", there are multiple outlets from which we can obtain images given that query. In particular, suppose we search on Google images and Twitter for "Golden Gate Bridge". As shown in Figure 1, most likely the results from Google will be stock images of the iconic bridge, usually taken from the same angle; on the other hand, popular Twitter images tend to look more diverse and ultimately more "personal". This discrepancy we empirically observe begs the following questions: How do photos that people post on social media differ from professionally taken photos of the same scene or event? And furthermore, are there features of the content that we can automatically learn, which differentiate it from "stock" content?

In this paper we set out to understand how and why those two types of image search results are different. Our overarching goal is to answer the following **research questions**:

1. Are there interpretable features that differentiate social images from stock images?
2. Are there interpretable features that social images and stock images share?

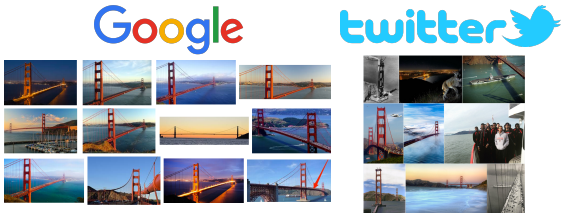

Figure 1: Google Image Search and Twitter Image Search results for "Golden Gate Bridge".

There exists recent work in the data mining community which attacks a similar question regarding primarily text-based web search results [1]. In particular, [1] demonstrates that highly retweeted Tweets can be used as a diverse and high-quality alternative to Google and Bing search. In this paper we aim to contrast social and web image search results using tensor analysis. Most relevant to the problem we set out to explore is the work of [8], which shows that transfer learning between the two types of content is not very successful. However, this is done in a supervised setting, whereas we propose to explore the problem without explicitly leveraging labels. This short paper carves out our vision for solving the problem. To the best of our knowledge, this is the first work to investigate the proposed application and research problems associated to it.

2. Proposed Method

2.1. Preliminaries

A tensor is a multi-dimensional extension of a matrix. The number of variables that index a tensor is

the order or the number of "modes" of a tensors. We denote a tensor as a boldface, uppercase, calligraphic letter such as $\mathcal{X}$. A matrix is denoted as $\mathbf{A}$ and a scalar as A or a. For indexing a matrix or a tensor we adopt Matlab notation, i.e., $\mathbf{A}(:, j)$ is indexing the j-th column of matrix $\mathbf{A}$. The outer product between two or more vectors is denoted by $\circ$.

2.2. Tensor Formulation

We propose to model the problem using tensors. In particular, suppose we are dealing with a single image search engine, e.g., Google Image Search. Suppose, further, that we have a set of queries of interest, and for each one of those queries we take the top-1 search result. The resulting images are likely to be of different sizes and resolutions, therefore, as a first approximation, we normalize all images to be 128×128 and grayscale. Subsequently, we vectorize each image by stacking every column of its pixel matrix on top of each other into a 16384-dimensional vector. We, thus, have a (query, top-1 image result) matrix for Google Image Search.

Depending on the search query, the temporal information of a particular result is potentially of crucial interest. Queries that pertain to topics of timely interest (e.g., "election" during October-November 2016) may offer widely different results across different days, or even for every hour of the day. Therefore, it is important to encode temporal information for every result. Assuming that we have collected results for all queries over a certain period of time, we have a tensor of (query, top-1 image result, time) for the Google Image Search results.

Our ultimate goal, however, is to contrast and compare Google Image Search to social media based search. For that, we need to collect a similar (query, top-1 image result, time) for a social media image search engine, such as Twitter image search. Those two three-mode tensors we have can be combined into a single four-mode tensor of (query, top-1 image result, time, search engine).

Given the above tensor, henceforth denoted as $\mathcal{X}$, there are multiple models we can use to analyze it [6]. In this work we use the CANDECOMP/PARAFAC or CP model [4, 5], which decomposes a tensor into a sum of $\mathbf{R}$ rank-one tensors:

$$\mathcal{X} \approx \sum_{r=1}^{R} \mathbf{A}(:, r) \circ \mathbf{B}(:, r) \circ \mathbf{C}(:, r) \circ \mathbf{D}(:, r).$$

In fact, we use CP with non-negativity constraints on the latent factors, a variant of CP that has been very popular and widely used in a variety of fields [10]. The motivation behind the non-negativity constraints, as in the original work introducing the Non-negative Matrix Factorization [7], is the fact that negative values in the factors do not hold physical meaning, since the images are better represented as a sum-of-parts. The objective function for CP with non-negativity constraints (often referred to as Non-negative Tensor Factorization) is:

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D}} \|\mathcal{X} - \sum_{r=1}^{R} \mathbf{A}(:, r) \circ \mathbf{B}(:, r) \circ \mathbf{C}(:, r) \circ \mathbf{D}(:, r)\|_F^2$$
$$\text{subject to } \mathbf{A} \geq 0, \mathbf{B} \geq 0, \mathbf{C} \geq, \mathbf{D} \geq 0,$$

where the inequalities are interpreted element-wise.

The reason why we choose CP over the multitude of tensor models that exist is twofold. First, CP is provably unique (up to permutations and scaling of the latent factors), which means that given a computed decomposition, the resulting factors are the only set of factors that can generate that particular solution, and no rotated version thereof. This, further, implies that we can safely interpret those factors, since they are unique. Second, the CP factors admit an immediate and intuitive interpretation. In particular, each rank one component of CP corresponds to a latent "concept" in the data; each factor matrix is an embedding of each tensor mode to the latent concept space:

- **Matrix A** is the query embedding. Every column of $\mathbf{A}$ is a soft-clustering assignment of queries to latent concepts, effectively bringing queries that yield pictorially similar results together.
- **Matrix B** contains the pixel embeddings. Each column of $\mathbf{B}$ after properly reorganized into a 128×128 image is the latent pictorial representation of that particular concept. This representation puts together elements of images that bear (sometimes partial) similarity.
- **Matrix C** contains the temporal profile of each latent concept. In other words, each column of $\mathbf{C}$ is a time-series whose intensity is proportional to the presence of a particular latent concept in the search results. We are particularly interested in $\mathbf{C}$ when analyzing newsworthy or trending image queries.
- **Matrix D** is the search engine embedding, showing how different search engines cluster into latent concepts, i.e., how much the search results of a particular search engine contribute to a latent concept. Therefore, each row of $\mathbf{D}$ is an assignment of search engines to a particular latent concept. If that assignment is of about equal proportions for every search engine, this implies that those search engines produce similar results which pertain to the topic of the latent concept. If, on the other

hand, the assignment is skewed (in the extreme case, only one search engine has a non-zero value), then this implies that this latent concept is present only in that search engine's results. The idea of search engine embeddings was first introduced in [1], and here we adapt it to image search.

Given the CP factors of our tensor and the above guidelines, we can explore, contrast, and compare image search results from Google and Twitter towards answering the research questions we pose in the introduction.

2.3. Discussion

The above formulation is a first step towards answering our overarching research questions. Inevitably, every such attempt entails different modeling decisions. Here we discuss the most important ones:

Why vectorize the images? When we form the tensor, instead of using the pixel matrix for the images, we vectorize them. One may argue that this approach is not taking full advantage of the two-dimensional structure of the image, which may contain different spatial correlations. However, Appendix A of [12] provides a thorough discussion and evidence in favor of vectorization, essentially demonstrating that vectorization of the image retains the two-dimensional spatial information in the subspaces discovered by Principal Component Analysis (PCA). In our case, the CP decomposition does not act exactly as PCA, but it rather seeks to identify latent factors. Therefore, there may exist subtle differences in the effects of vectorization than the ones shown in [12] which we intent to investigate as future work.

Modeling top-k results? The above formulation takes into account only the top-1 result. However, as we allude in the introduction, the diversity of results even for a single query is potentially a distinguishing factor between web and social image search. In order to incorporate, say, the top-5 results we can add different modes to the tensor corresponding to the position of the result. For instance, we can form a (query, image result, time, search engine, top-k index) tensor, and we reserve investigation of this approach in future work.

Selecting the number of latent factors The quintessential question in all exploratory analysis tasks is how many latent concepts (clusters) should we choose. Answering this question, in particular for tensor analysis, is NP-hard, however there exist good heuristics in the literature. However, heuristics based on the core consistency [3, 9] are only applicable in the case where R is smaller than the smallest of the mode dimensions of the tensor. In our case, the smallest mode is the one corresponding to search engines (in all our examples and experiments it was 2), therefore [3, 9] are not applicable. This is a very challenging and exciting problem but is beyond the scope of this work. For the purposes of our exploration, we manually tune R.

3. Case Study

3.1. Data Collection

For the purposes of demonstrating a proof of concept for our approach we collect a small dataset from Google Image Search and Twitter. The list of queries we decided to focus on is shown in Table 1. Those queries were about topics of current interest (e.g., US Presidential Election and Hurricane Matthew) during the data collection that took place between October 8 - November 11, 2016. We stress that this is a small dataset used exclusively for demonstrating the key idea of this paper. In follow-up work we plan to collect and analyze a larger (in duration and number of queries) dataset. For Google image search we used the Google Custom Search API, and for Twitter image search we used the Twitter REST API and the Tweepy Library. Both for Google and Twitter we adopt the ranking that the respective search engines use. However, in future work we intend to experiment with different social signals, such as the number of retweets which [1] has demonstrated that can yield web search results of high quality.

3.2. Results

We use the Tensor Toolbox for Matlab [2] and in particular the `cp_nmu` implementation of the Non-negative Tensor Factorization.

We manually experiment for the selection of R by inspecting the results. The results we present come from a set of results with $R = 40$ and were hand-picked, having the clarity of the latent image as primary selection criterion. For each latent component we create a subfigure that contains (in clock-wise order): (a) the latent image as captured by the corresponding column of $\mathbf{B}$, (b) the top queries for that component as indicated by the largest values of the corresponding column of $\mathbf{A}$, (c) the temporal profile of the component as captured by matrix $\mathbf{C}$, and (d) a scatterplot with a single point whose coordinates indicate the participation of Twitter and Google to the particular latent component, from matrix $\mathbf{D}$.

In Figure 2 we show six components that pertain to topics such as 'Election", "Refugee", and "Donald Trump", and in Figure 3 we show two results of a popular artist, Bruno Mars. We make the following observation on our preliminary results:

Table 1: List of queries

Election	Gun control	Refugee	Artificial Intelligence
Hurricane	Donald Trump	Mark Zuckerberg	Bruno Mars
Lakers	Ford	Jimmy Kimmel	Los Angeles

Figure 2: Each sub-figure contains a summary of a CP component. In clock-wise order: (a) the latent image, (b) the top queries for that component (note that their scores do not need to sum up to 1) (c) the temporal profile of the component, and (d) a scatterplot with a single point whose coordinates indicate the participation of Twitter and Google to the particular latent component. The shown components mostly involve queries such as "Election", "Refugee", and "Donald trump".

Figure 3: Two different CP components for "Bruno Mars".

- The latent images for Google-dominant components usually resemble stock images. On the other hand, Twitter-dominant components have latent pictures that are more emotionally loaded: for instance, the crib in the latent image pertaining the the "Refugee" query, the two children in the "Hurricane" latent component, and latent image for the "Donald Trump" and "Election" component, that contains elements of a meme which spells "Russia First". This observation confirms the empirical intuition of the introduction.
- Twitter-dominant components tend to involve more queries with significant participation than Google-dominant ones which tend to be more clean-cut. This may be attributed to the inherent high diversity of socially popular images.
- In the case of Bruno Mars query, we observe that when there is a (even small) Twitter influence, the latent component contains timely elements, such as the advertisement for the singer's new tour, instead of the singer's face which is prevalent in Google-dominant components.
- With respect to the temporal evolution of different latent patterns, we observe that for components that are made up by primarily stock content (such as the top-left component of Figure 2) their temporal pattern is mostly stable throughout our data collection period. To the contrary, components that are made up by social content (such as the rightmost components of the second and third rows of Figure 2 referring to Russia's potential connection with the 2016 Presidential Elections and Hurricane Matthew respectively) are more bursty in time.

4. Conclusions & Future Work

This paper outlined our vision on exploring the similarities and differences between web and social image search. We propose a tensor-based analytic framework that is able to offer interpretable results, and we provide a proof of concept by analyzing a dataset we collected for this purpose. Furthermore, this paper outlines a set of interesting and exciting future research directions:

1. What constraints, other than non-negativity, for the CP decomposition are well suited for this application and have potential to offer more interpretable results?
2. Can we use other tensor models in order to extract more complex, yet interpretable, features (e.g., the Tucker decomposition used in the popular TensorFaces method [11])?
3. How can we incorporate additional meta-data, such as Twitter retweets, to enhance the model's understanding of the underlying latent concepts?

5. Acknowledgements

This work was supported by the Bourns College of Engineering at the University of California Riverside. The authors would like to thank the anonymous reviewers, and especially Reviewer # 3 for their incredibly constructive comments and suggestions. Finally, the authors would like to thank Rakesh Agrawal whose collaboration with E. Papalexakis [1] served as inspiration for this paper.

References

[1] R. Agrawal, B. Golshan, and Evangelos E. Papalexakis. Whither social networks for web search? In *ACM KDD'15*. 1, 3, 5

[2] B. W. Bader, T. G. Kolda, et al. Matlab tensor toolbox version 2.6. Available online, February 2015. 3

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

[4] J. D. Carroll and J.-J. Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition. *Psychometrika*, 35(3):283–319, 1970. 2

[5] R. Harshman. Foundations of the parafac procedure: Models and conditions for an" explanatory" multimodal factor analysis. 1970. 2

[6] T. Kolda and B. Bader. Tensor decompositions and applications. *SIAM review*, 51(3), 2009. 2

[7] D. Lee and H. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999. 2

[8] N. C. Mithun, R. Panda, and A. K. Roy-Chowdhury. Generating diverse image datasets with limited labeling. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 566–570. ACM, 2016. 1

[9] Papalexakis, Evangelos E. Automatic unsupervised tensor mining with quality assessment. In *SIAM SDM*, 2016. 3

[10] A. Shashua and T. Hazan. Non-negative tensor factorization with applications to statistics and computer vision. In *Proceedings of the 22nd international conference on Machine learning*, pages 792–799. ACM, 2005. 2

[11] M. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. *Computer Vision ECCV 2002*, pages 447–460, 2002. 5

[12] M. A. O. Vasilescu. *A multilinear (tensor) algebraic framework for computer graphics, computer vision, and machine learning.* PhD thesis, Citeseer, 2009. 3