Finding Image Exemplars Using Fast Sparse Affinity Propagation

Yangqing Jia†, Jingdong Wang‡, Changshui Zhang†, Xian-Sheng Hua‡
†State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
Department of Automation, Tsinghua University, Beijing, China
‡Internet Media Group, Microsoft Research Asia, Beijing, China
†jiayq06@mails.thu.edu.cn, ‡{i-jingdw,xshua}@microsoft.com, †zcs@mail.thu.edu.cn

ABSTRACT

In this paper, we propose a novel approach to organize image search results obtained from state-of-the-art image search engines in order to improve user experience. We aim to discover exemplars from search results and simultaneously group the images. The exemplars are delivered to the user as a summary of search results instead of the large amount of unorganized images. This gives the user a brief overview of search results with a small amount of images, and helps the user to further find the images of interest. We adopt the idea of affinity propagation and design a fast sparse affinity propagation algorithm to find exemplars that best represent the image search results. Experiments on real-world data demonstrate the effectiveness of our method both visually and quantitatively.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval; I.4.0 [**Computing Methodologies**]: Image Processing and Computer Vision

General Terms

Algorithms

Keywords

Image Exemplar, Web Search, Sparse Affinity Propagation

1. INTRODUCTION

With the explosive development of the Internet technology, we are able to find a large amount of information via simple queries using state-of-the-art search engines such as Google, Yahoo, Live search, etc. Recent years have witnessed a quick development of the text/document search, and as recent developments in digital media technology have made the production of digital images much easier and more popular, efficient image search have become an interesting issue in the multimedia research area. A lot of attention has been paid to more accurate image searching, such as performing PageRank [6] on images to search for most relevant images the user is interested in, or adopting both visual and textual information [3].

An important issue independent from image searching algorithms is that, assuming the image search results to be given (which often involves a large number of images), how can we efficiently present them to the user? Traditionally, most search engines return a ranked list of images, which may ramble on for tens of web pages, leading to an exhaustive user experience. To improve this, Some works on clustering images such as [1, 5] have been proposed to improve user experience by organizing image search results into groups each corresponding to a different view of the search result. However, these works are mainly based on the classical clustering algorithms, and although they can produce a more organized result, how to present the clusters to the user is still an open problem since the number of images in each cluster may still be large.

In this paper, instead of simple clustering, we aim to discover representative images, which we call "exemplars", from search results and simultaneously cluster the images into groups characterized by these exemplars. The exemplars are then delivered to the user in a single search result page (usually containing about 20 images) that is much easier to browse. Similar thought has also been used in scene summarization [7]. Specifically, an exemplar should be as similar as possible to the other images in the cluster it represents, and exemplars should be as diverse to each other as possible so that they carry little redundancy. To cluster the data and find the exemplars simultaneously, we adopt the affinity propagation (AP) idea. Also, considering the real-time requirement of the search application, we will derive a fast sparse AP algorithm that runs much faster than the original AP algorithm.

In the following parts of the paper, we will first theoretically introduce the fast sparse affinity propagation algorithm, and then apply it to the image search result clustering application to show the efficiency and effectiveness of the method.

MM'08, October 26–31, 2008, Vancouver, British Columbia, Canada.

2. FAST SPARSE AFFINITY PROPAGATION

2.1 Notations

Formally, given a set of n data points $\mathcal{X} = \{x_1, x_2, \cdots, x_n\}$

and the similarity measure $s(x_i, x_j)$ between two data points, we aim to cluster the data into m ($m < n$) clusters, each represented by an "exemplar" from $\mathcal{X}$. These exemplars form a subset of the original data set as $\mathcal{X}_e = \{x_{e1}, x_{e2}, \cdots, x_{em}\} \subset \mathcal{X}$. Denote the exemplar of each data point x in $\mathcal{X}$ by $e(x)$, we aim to maximize the sum of similarities between each data and its exemplar as follows:

$$S(\mathcal{X}, \mathcal{X}_e) = \sum_{i=1}^{n} s(x_i, e(x_i)). \tag{1}$$

The difference between finding exemplars and classical clustering methods such as k-means is that, in classical clustering problems, the centroid for each cluster may not necessarily be a real data point (for example, in k-means, the centroid is the mean of the data points in each cluster). These centroids may not have a real-world interpretation. For example, the mean of several images may not be a plausible image. However, in the image search result clustering problem, as we need to present each cluster to the user, it is necessary to find an exemplar for each cluster instead of those centroids that lack real-world meanings.

Finding exemplars has been an interesting issue in the computational society. The most popular way may be the K-centers algorithm such as [4]. Many of the algorithms run in a greedy way to find K sub-optimal exemplars that represents the data. Recently, Frey et al. proposed affinity propagation to find exemplars from a set of data [2]. It has been proved to be more effective than the classical methods. We will briefly review the AP algorithm, consider its improvements, and apply it on the image search result clustering problem.

2.2 Affinity Propagation

In brief, the AP algorithm propagates two kinds of information between each two data points [2]: the "responsibility" $r(i, k)$ sent from data point i to data point k, which reflects how well k serves as the exemplar of i considering other potential exemplars for i, and the "availability" $a(i, k)$ sent from data point k to data point i, which reflects how appropriate i chooses k as its exemplar considering other potential points that may choose k as their exemplar. The information are updated in an iterative way as

$$\begin{aligned} r(i,k) &:= s(x_i, x_k) - \max_{k' \neq k}\{a(i,k') + s(x_i, x_{k'})\}, \\ a(i,k) &:= \min\{0, r(k,k) + \sum_{i' \notin \{i,k\}} \max\{0, r(i',k)\}\}. \end{aligned} \tag{2}$$

The self-availability is updated in a slightly different way as

$$a(k,k) := \sum_{i' \neq k} \max\{0, r(i',k)\}. \tag{3}$$

Upon convergence, the exemplar for each data point x_i is chosen as $e(x_i) = x_k$ where k maximizes the following criterion:

$$\arg\max_k \quad a(i,k) + r(i,k). \tag{4}$$

The justification of the AP algorithm roots from the max-sum algorithm in a factor graph constructed from the data. We refer to the supplement materials of [2] for detailed discussions. Although AP still does not guarantee global optimum, several experiments in [2] have shown its consistent superiority over the previous algorithms.

The original AP algorithm takes the full similarity matrix to perform propagation. In each stage, there are generally n^2 data pairs whose responsibility and availability values need to be calculated, resulting in a computation complexity of $O(n^2T)$ where T is the number of iterations. This greatly affects the speed of the algorithm especially when the number of data points is large. On the other hand, when the data are sparsely related because the similarity between some data points are unknown or be $-\infty$, the affinity propagation algorithm can be adjusted to utilize the sparsity. This is achieved by constructing a sparse graph structure $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$, where the vertices are data points and the edges $\mathcal{E}$ contains parts of the pairwise edges between any two of the data points. It has been pointed out in [2] that the sparsity will lead to faster calculation since the propagation of responsibility and availability only needs to be performed on the existing edges.

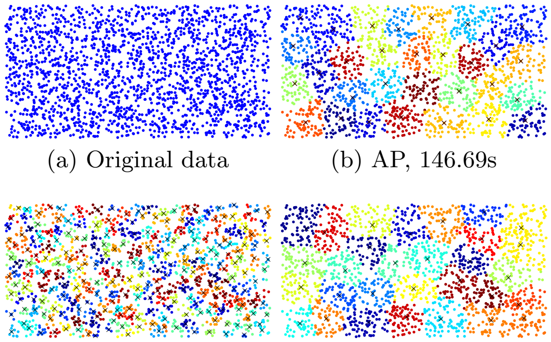
One may easily raise the question: if the sparsity is able to boost the speed, can we actively create a sparse graph to perform affinity propagation and to minimize the deterioration of the sum-of-similarity measure from the non-sparse case? However, to the best of our knowledge, the sparsity has up to now been considered only as a property of certain problems, and there has not been any method that actively utilizes sparsity to speed up the propagation procedure when the full similarity matrix is obtainable. In the following part, we will propose a new way to construct a sparse graph and perform affinity propagation faster than the classical AP algorithm.

2.3 Two-stage Fast Sparse AP

The key point to benefit from both the effectiveness of the AP algorithm and the sparsity of the similarity matrix is to choose the appropriate edges between data points. Our thought is based on two aspects: (1) for two data points that are far apart, if we pre-assume that neither of them chooses the other as exemplar, whether to add an edge between the two data points does not change the final result. Thus the propagation algorithm may be boosted if we are able to discard these edges; (2) data points that serves as good exemplars locally may be candidates for exemplars globally. Based on these, our method mainly consists of two stages: sparse graph construction and iterative edge refinement.

Sparse Graph Construction. we first construct a sparse graph structure $\langle \mathcal{V}, \mathcal{E} \rangle$ similar to the K-nearest neighbor approach: if point x_i is among the K points that has the largest similarity with point x_j, then x_i, x_j are connected by an edge (i, j). This generally leads the edge set $\mathcal{E}$ to have roughly Kn edges (as the K-nearest neighbors are not symmetric, there may be more than K edges connecting to each data point x_i). Usually, we choose the parameter K to be much smaller than the number of data points. This leads to a very sparse graph, and propagation on it only takes computation complexity of $O(nT_1)$ where T_1 is the number of iterations AP takes. Although the number of iterations is difficult to represent analytically, this complexity is generally linear with respect to n.

The problem with AP on a sparse graph is that, for a data point with K neighbors, it can and only can be exemplar of $K + 1$ data points (its neighbors and itself). Thus for a K-nearest neighbor graph, there are at least n/K exemplars, which is much more than expected. Thus, we will consider refining the edges to reduce the number.



(a) Original data (b) AP, 146.69s

(c) FSAP Stage 1, 10.53s (d) FSAP Stage 2, 11.99s

Figure 1: Toy 2-D data. "×" indicate exemplars and colors indicate clusters.

Iterative Edge Refinement. We consider the exemplars from the first stage as the candidates for final exemplars, and consider the other data points as non-exemplars. A new graph structure is constructed who adopts three criteria to determine edges:

(1) If candidate x_i is the exemplar of x_k, then x_i, x_k are connected by an edge (x_i, x_j).

(2) For two candidates x_i, x_j, if there exists two data points x_k, x_l that satisfy $s(x_k) = x_i, s(x_l) = x_j$ (i.e., they take x_i, x_j as their exemplar respectively) and are K-nearest neighbor to each other, then x_i, x_j are connected by an edge;

(3) For two candidates x_i, x_j, if they are connected by criterion 2, then all data points that choose x_i as exemplar are connected to x_j, and vise versa.

Affinity propagation on this graph is able produce fewer exemplars than the previous graph. Multiple exemplar candidates may merge into one group because nearby candidate and the corresponding data points are connected by criterion 3. Also, the graph constructed in this way preserves the advantage of sparsity: The edges between candidates are sparse because only nearby ones are connected; also, there are no edges connecting two data points if neither are candidates; for non-candidate points, they are only connected to its exemplar candidate in the previous stage and the nearby few candidates. It is difficult to analytically calculate how many edges there are exactly. However, empirically we find that the edges are roughly proportional to n, thus the computational complexity is about $O(nT_2)$ where T_2 is the number of iterations AP takes. This complexity is also significantly lower than the original AP algorithm.

For data sets containing large number of data points, the second stage may be iteratively performed so that the number of exemplars is reduced to a desirable value. In our experiments where data sets contain about 1000 data points, only one iteration is needed to find the desired number of exemplars.

We adopt a simple toy data set to show the efficiency of our method. 2000 data points are randomly sampled from a 2-D rectangle as shown in Fig. 1(a), and we aim to find 30 exemplars among them. AP using full similarity matrix (the similarity between two data points are calculated as the negative distance between them) took 146.69 seconds to produce the final result as shown in Fig. 1(b). Using our FSAP approach with initial neighborhood $K = 15$, we find 252 candidates in the first stage in 10.53 seconds, and 30 final exemplars in the second stage in 11.99 seconds, shown respectively in Figs. 1(c) and 1(d). In general, our methods takes only 15% of the processing time and achieves competitive result against the original AP algorithm. The advantage on computation time can be also verified in Fig. 3(a) where the number of data points varies from 500 to 3500. It can be seen that the time of FSAP is approximately linear to the number of data points while AP takes much more time to compute.

(a) (b)

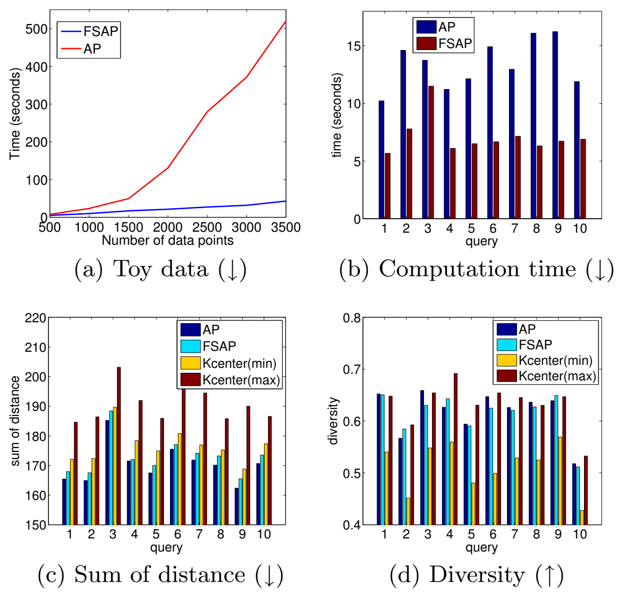
(c) (d)

Figure 2: Visual results corresponding to query "Flowers", showing four exemplars and several other images represented by the exemplars.

3. EXPERIMENTAL RESULTS

We use several real-world query results to show the potential of our method. For data preparation, we used the Google image search to find images with ten different query strings the same with [3] and crawled the first 1000 of each query's returned images. Instead of the data such as Mona Lisa used in e.g. [6], these images have a large diversity and covers a wide range of appearances. Thus, we adopt the color moments of each image's HSV channel as the features to capture the visual characteristics of each image. The similarity between two images is considered as the negative distance between their feature vectors. The neighborhood number K is identically set to 15.

Figure 2 shows a typical result on the images obtained from query "Flowers". Due to limit of space we only provide four exemplars and several other images that are represented by each exemplar. It can be seen that these exemplars represent each cluster well: they show (a) images containing small flowers, (b) pictures of flowers on a black background, (c) brightly colored close-up shots, and (d) illustrations on a white background, respectively. If a user wants to find close-up shots that has vivid colors, the exemplars may lead to the right subset c, eliminating other images such as (b) or (d) that s/he may not be interested in. Figure 4 shows several exemplars found for other queries.



(a) Toy data (↓) (b) Computation time (↓)

(c) Sum of distance (↓) (d) Diversity (↑)

Figure 3: Quantitative results: (a) the computational time of AP and FSAP on the toy data when the number of data points varies; (b) the time used to cluster each query; (c)(d) the sum of distance and diversity value for each query. The arrows indicate whether larger (↑) or smaller (↓) value is better.

To quantitatively evaluate the performance, we compare our method with the affinity propagation algorithm using full similarity matrix, and adopt the classical k-center algorithm [4] as the baseline. Three criteria, including computation time, representative and diversity scores of the exemplars, are adopted for quantitative comparison. The comparison results are shown in Fig. 3(b)-(d). The sum of distances between data points and their exemplars, which is the negative of Eqn. 1, is used to evaluate how well exemplars represent other images. The median value of distance between exemplars is used to evaluate the diversity of exemplars. We run the k-center algorithm 100 times and report the best and worst result here. We also compared the computational time of the classical AP and our FSAP method. The results are shown in Fig. 3. It can be seen that our method achieves competitive representative and diversity scores compared with AP algorithm using full matrix, but takes significantly less time. In the meantime, both of them perform competitively against even the best result of the k-center algorithm, showing the effectiveness of the affinity propagation approach.

It is worth pointing out that, although we do not adopt other information such as surrounding textual feature [3] for images, they can be naturally embedded in our method by modifying the similarity measure so that more information is considered. However, as our paper mainly focuses on the exemplar finding method itself, detailed discussion on choosing features is beyond the scope of the paper.

4. CONCLUSION

How to present image search results to the user is an interesting issue in real-world applications. In this paper, we proposed a new clustering method based on fast sparse affinity propagation to enhance the presentation. We aim to cluster the search into several groups each represented by an exemplar, and by adopting affinity propagation, our method is able to find a set of exemplars that well represents the large amount of images within a reasonable time. This gives us possibility to show the user a few representative images instead of a large number of unorganized image search results and can improve the user experience.

In the future work, we are planning to incorporate more information such as the textual data and image features other than color moments to better characterize the similarity between two images, so the clustering may be more accurate and have better semantic interpretation. We also plan to perform user-experience based evaluation to more accurately test the performance of our method.

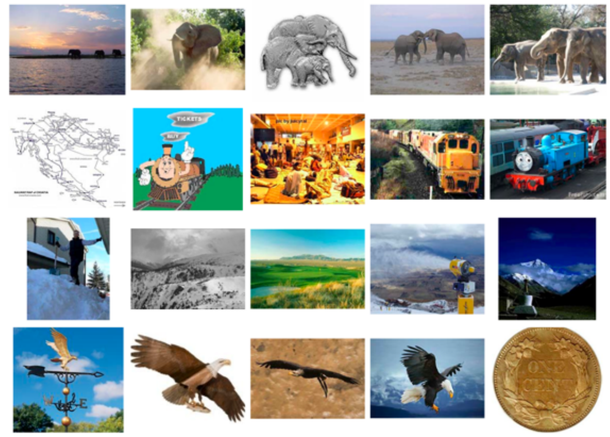


Figure 4: Representative exemplars, each row corresponding to queries "Elephants", "Railways", "SnowMountain", and "FlyingEagle" respectively.

5. REFERENCES

[1] D. Cai, X. He, Z. Li, W. Ma, and J. Wen. Hierarchical clustering of WWW image search results using visual, textual and link information. *Proc. of ACM MM'05*, pages 952–959, 2004.

[2] B. Frey and D. Dueck. Clustering by Passing Messages Between Data Points. *Science*, 315(5814):972, 2007.

[3] B. Gao, T. Liu, T. Qin, X. Zheng, Q. Cheng, and W. Ma. Web image clustering by consistent utilization of visual features and surrounding texts. *Proc. of ACM MM'05*, pages 112–121, 2005.

[4] T. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38(2):293–306, 1985.

[5] Y. Hu, N. Yu, Z. Li, and M. Li. Image Search Result Clustering and Re-Ranking via Partial Grouping. *Proc. of ICME'07*, pages 603–606, 2007.

[6] Y. Jing and S. Baluja. Pagerank for product image search. In *Proc. of ACM WWW'08*, pages 307–316, 2008.

[7] I. Simon, N. Snavely, and S. Seitz. Scene Summarization for Online Image Collections. *Proc. of ICCV 2007*, pages 1–8, 2007.