Unsupervised Image clustering

Ashish Agrawal
Department of Computer Science and Engineering
in the guidance of
Prof. Harish Karnick
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

April 20, 2009

Abstract

Extracting semantic information from images has attracted much attention in the domain of computer vision and image processing. Areas like face recognition, detection, tracking etc. work on identifying semantics in images. In this paper we attempt to cluster images based on their semantic content. The approach involves segmenting the image at different scales and extracting interesting patches in the image. A over sized dictionary of these patches is constructed. Every image is then back projected on this dictionary to obtain a sparse feature vector for the image. This is then used to cluster the image. If we label the clusters, a new image can also be assigned to one of the existing clusters to achieve a classification. Unsupervised clustering has wide applications in information retrieval and image search. The algorithm can be possibly used to filter image search results. The paper describe the algorithm, the various techniques used and the results obtained.
1 Introduction

Every image contains semantic information which is what we infer after seeing the image. This semantic information could be the presence/absence of a certain set of objects or the attributes of these objects or their relative positions with respect to each other. Understanding the semantics in an image has great applications in areas ranging from face detection, medical imaging to image retrieval. With the growing volume of multimedia content in the internet the effective organisation of this content becomes a task of paramount importance. Clustering is one of the possible approaches to effective organization and retrieval of images. In a simplistic statement, clustering is the division of a set of observations into groups based on some predefined criterion and a similarity measure. In our case this criterion is the semantic information in the image which is characterised by the presence or absence of distinguishing image patches in the image. Hence we set on defining a similarity measure between images and consequently clustering images based on this measure.

2 Previous Work and Literature Review

Image processing can be broadly divided into three categories. -

- Low level image processing
- Mid level image processing
- High level image processing

Low level image processing deals with tasks like contrast management, thresholding, noise removal/addition etc. These are quite local in nature and only consider an environment of a few pixels in doing the operations. Mid level processing includes segmentation, edge detection etc. which require a bigger environment to be taken into consideration and is higher up in the value chain. High level image processing deals with identifying the contents of the image. This includes techniques like face detection, recognition, tracking etc. Clustering is a high level image processing task but we also make use of the lower level routines like segmentation to achieve our objectives.

There is a enormous amount of literature dealing with image processing. A significant portion deals with low level and mid level processing[1]. There has also been significant research in computer vision areas like face recognition and detection, tracking etc. However, much of the research in high level image processing and computer vision has been of the supervised nature. This could possibly be because of the availability / feasibility of arranging large sets of labelled data or possibly the need for a high level of accuracy. Web image management and retrieval however is one area where there is a significant dearth of labelled data sets and a moderate level of accuracy is permitted. It is in this area where unsupervised image analysis has a key role to play.

Previous research has attempted to construct feature vectors for images through a number of ways and then run classifiers or clustering algorithms on them. The techniques used include various filters, transforms, spatial analysis etc.[2] [3] [4] There have also been attempts to characterise standardised images in a specific environment by eigen faces.[5] Histogram analysis and more sophisticated techniques like Gaussian mixture modelling have also been used for constructing a similarity measure between images.[6] The latter approach has produced decent results for outdoor images. However, the reach of these techniques is rather limited and quite often they end up comparing just the texture, colour distribution in the image rather than semantic information[7][8][9]. There have been research based on the region of interest in an image as well.[10] The eigen faces approach works well in standardised cases, however it’s performance degrades when we have a more open image dataset.

In this paper we present our attempts to cluster images using Manifolds and using a segmentation with K-SVD approach.
3 Manifolds

Manifolds are subspaces of higher dimensional spaces. A manifold could correspond to a curve over a sphere, a line through a sphere etc. Manifolds can be used to identify pattern in high dimensional data. The ISOMAP algorithm, an implementation by Tanenbaum works on identifying a manifold through a k-nearest neighbour approach.

The assumption we start with when using the manifolds approach is that images with the same semantic content lie closer to each other across some distance measure. Hence they form a manifold of their own. The manifold we obtain for a set of images which are semantically similar should also have a lower dimensionality than the manifold we obtain for images which are not. The ISOMAP algorithm had previously been used to identify face orientations and cluster them and this provided with an inspiration to use the manifolds approach.[11][12] The L-2 norm between images was used to construct the euclidean distance measure between two neighbouring images. The algorithm was tested with nearest neighbour values of 3,5 and 7 on a dataset containing images of buildings and mountains. It was expected that the dimensionality of the manifold for any of the categories alone would be lesser than the dimensionality of the mixed category. With any manifold assignment to the category is associated an error whose convergence gave us an idea about the dimensionality of the manifold. We were however unable to validate our claims and the manifold algorithm did not converge any faster for a semantically similar category than for a mixed category. Hence we concluded that perhaps the L-2 norm is not a good measure of comparing images.

4 Segmentation

To understand the semantics in an image, it is important to identify the areas where the interesting semantics are present. This can be done using a segmentation algorithm. Segmentation essentially divides an image into areas which are similar inside and contrast with their adjacent area. The task of the algorithm was just to provide the future stage with the set of important patches for the image. The segmentation was accomplished using the OpenCv toolkit[13][14] with the Pyramid Segmentation routine. A multi-level segmentation was done where the image is reduced in size by half at every step to identify bigger and bigger paths in the image. Hence we are returned a set of patches which range from 2X2 pixels in size to the size of the image. This provides us with the ability to extract semantic information of any scale from the image. For a simplistic analysis all the patches were extracted as squares enclosing the segment.
5 K-SVD dictionary construction

The eigenfaces approach had a lot of merit where it tried to characterise a new face as a linear combination of the eigenfaces present in the dictionary.[15] This basically works on exploiting the semantic similarity between face images. The eigenfaces approach was also applied to non-face images with decent levels of accuracy.[16] The K-SVD approach was an inspired from the eigenfaces approach but aimed at eliminating the bottleneck of the eigenfaces approach of requiring standardised images.[17][18][19] Singular Value Decomposition (SVD) is an important factorization of a rectangular real or complex matrix. The K-SVD is an extension of the K-means and SVD algorithm where a dictionary is constructed by K applications of the SVD algorithm. The K-SVD algorithm intends to construct a dictionary for every patch size. The dictionary is a set of vectors which can be linearly combined to produce a patch of the given size. There are two main aspects of the dictionary -

- The dictionary is oversized
- The dictionary is constructed in a way so as to reduce the number of vectors required to represent a patch. It aims for a sparse representation

The oversized dictionary is useful for generalising the eigenfaces approach to a more robust environment while the sparse representation is useful in the eventual clustering. The patches were upsized to the nearest power of 2 so as to limit the number of dictionaries we need to consider. Hence we had dictionaries of patch sizes 2X2, 4X4, 8X8, 16X16 and so on. Only the patches returned by the segmentation algorithm were considered for the purpose of training the dictionary.

If we have a patch size of dXd, then in principle we can represent every patch of this size with a linear combination of d*d patches. So dXd forms a lower bound on the number of images one would have in the dictionary. To make the dictionary oversized, we introduce a redundancy factor r which leads to r*d*d patches. The dictionaries obtained for patch sizes of 8X8 are being shown here both with a redundancy ratio of 1 and 4. The K-SVD computes the dictionary D corresponding to -

\[
\text{argmin} ||\alpha||_0 \text{ subject to } ||D\alpha - x||^2 < \epsilon
\]

6 Back-projection and image feature vector

Once the dictionary has been obtained, we back project every patch we had received from the segmentation routine and obtain a corresponding weight vector for it which is of the size of the dictionary. If the image
is $I$, feature vector $v$ and dictionary $D$, we minimize $||v||_0$ subject to $||Dv - I||^2 < \epsilon$. Hence we try for as sparse a representation as possible. The weights over all the similar sized patches of the image are summed over and this is done for all patch sizes. Hence we obtain a sparse feature vector of the combined sizes of all the dictionaries for the image. With a redundancy ratio of 4 and the patch size limit being 8x8 we obtain a feature vector of size 336.

7 Clustering

Once we obtain a feature vector for every image, the dataset is clustered using one of the standard clustering techniques like K-means. We implemented K-means using the MATLAB toolkit. Since it would be the case that some dictionary elements would be more distinguishing for clustering purposes than others, a weighted approach based on feature selection would probably be useful here.

8 Results

The algorithm was run on a dataset of landscape images as well as on the Amsterdam Library of Small objects. In both cases we got definite semantic bias in the clustering we obtained. However, we were not able to give a quantitative estimate of the accuracy of the algorithm due to the subjectivity of an ideal clustering. The algorithm needs to be tested on datasets which have a definite semantic division in them to get a better estimate of its accuracy.

9 Future Work

The algorithm needs to be tested on an objective dataset to get a better estimate of its accuracy. As of now the algorithm only considers square patches. This needs to be expanded on cover rectangular patches and contours as well. As of now we are not considering the geometry of the patches. We also need to include a measure based on the relative geometric distance between patches. This would help us distinguish between a sunset and a noon scene. Finally, the implementation needs to be made more efficient to deal with larger patch sizes.
References


