CS164 Project Final Write-up

Max-Margin Content Based Image Search

Mohammad Rastegari

November, 22th 2010

Abstract

Nowadays in the internet there are many web-pages containing both images and texts. However, almost all search engines are based on textual information, more often humans surf in web based on images that they see in web-pages. In other words, they can find the main context of a web-page using images in that web-page. Images and texts share some information about any context. Here there is an important question: *Which of them describe the other?*. Normally we may think that texts are describing every things but in practice we first see pictures then think about texts. However, sometimes we may see the pictures but we still can not understand any thing about the context then we have to read the texts which are related to the pictures. In this project I tried to devise a method to find out if a text and an image are semantically related to each other or not. It can help web search engines to retrieve web pages containing both texts and images related to the users' query. I found a Max-Margin formulation for the problem that leads to a SVM learning.

Introduction and Literature Review

Conventional image search methods were based on the textual annotations on the image files. In other words, the image search was text search rather than a real image search. These methods result in a drawback. If the annotations of a particular image do not correctly describe the image, retrieved images will not be the images that user expected to see.

Recently some researches[2] have been performed in order to find the near-duplicate images of a given image. You can see the "Find similar image" by Google. In these methods they try to find a fast algorithm to search for similar images in terms of their appearance features. But we are interested in having an image search that retrieves the images that have same context(*not appearance*) or come from the same category as the query image or text. In [3] an image search has been designed which retrieves the images almost from the same category of the query image. The main difficulty of this problem is dealing with unseen categories, which are images belong to the categories that we did not have them in training data.

Image categorization is not always a matter of learning methods, it could be a matter of designing suitable descriptors. The conventional image descriptors were not suitable for categorization task. In [4] a descriptor has been designed that shows very good results on image categorization problems.

In [1] the authors proposed a method to describe an image by a sentence or vice-versa. They create a semantic space (or by their word *meaning space*). Meaning space constructed by a triple {*Object, Scene, Action*}. For a given text they (Farhadi et al [1].) tried to find such a triple using NLP methods. For every image they tried to match a corresponding triple using object detection, scene discovery and action recognition methods. Then if an image and a text are near to each other in the meaning space, they could have same semantic. In this project I focused on the problem of matching between image and text(sentence). Ideally an image should be matched with a sentence which is semantically related to it. *I did not design a fast algorithm.* I designed a method which is able to automatically learn a semantic space using the aligned image and text features as training data. In next part I will formally explain my proposed method.

Bilinear Model

Suppose that we have a set of images S_{img} and a set of text S_{txt} such that each text can be used as an explanation of one or more images and

each image has same semantic as one or more texts. We show this by an assignment matrix M.

$$M = \begin{bmatrix} +1 & -1 & \dots & +1 \\ -1 & +1 & \dots & -1 \\ \vdots & \ddots & \vdots & \\ +1 & -1 & \dots & +1 \end{bmatrix}$$

columns in M are text indices and rows are image indices. The element M_{ij} is equal to $+1$ when the j^{th} text is a meaningful explanation of the i^{th} image. Otherwise it is -1. We are interested in finding a matrix W which maximize the following optimization problem.

$$W = \underset{W}{argmax}\{\sum_{\forall ij} ({I_i}^T \times W \times T_j) \times M_{ij}\}$$

I_i is the i^{th} image feature vector and T_j is the j^{th} text feature vector. We can reformulate the above equation as follows:

$$\widetilde{W} = argmax\{\sum_{\forall ij} {F^{ij}}^T \times \widetilde{W} \times M_{ij}\}$$

$$\widetilde{W} = \begin{bmatrix} W_{11} \\ W_{12} \\ \vdots \\ W_{kl} \\ \vdots \\ W_{mn} \end{bmatrix} \quad and \quad F^{ij} = \begin{bmatrix} I_i(1)T_j(1) \\ I_i(1)T_j(2) \\ \vdots \\ I_i(k)T_j(l) \\ \vdots \\ I_i(m)T_j(n) \end{bmatrix}$$

Now we can easily devise an optimization problem in the Max-Margin way.

$$min\|\widetilde{W}\| \quad subject\ to: \ \forall ij\{{F^{ij}}^T \times \widetilde{W} \times M_{ij} \geq 1\}$$

Which is a simple SVM.
In training we use several pairs of image and text $\{Image_t, Text_t\}$ to learn the W and in testing we use different pairs of image and text to predict their labels as follows:

$$C = {I_t}^T \times W \times T_t \tag{1}$$

C will be the confidence measure that tells us how much the text T_t is semantically similar to the image I_t. When C is a large positive value

it means high similarity and when is large negative value it means high dissimilarity.

So far we found a method that enables us to find out if a given pair of text and image are semantically similar to each other or not. Now lets consider the application that we are given an image as a query and we would like to retrieve other images which have similar semantics as the given query image. In order to solve this problem we need to find a way to map every image into a semantic space then in that space we can find some nearest neighbours as a result. Lets decompose the matrix W by SVD decomposition.

$$W = S\Sigma V^T$$

Now we can rewrite the W by the multiplication of two matrixes A and B.

$$W = S\Sigma^{\frac{1}{2}}\Sigma^{\frac{1}{2}}V^T$$

$$A = S\Sigma^{\frac{1}{2}}$$

$$B = \Sigma^{\frac{1}{2}}V^T$$

$$W = AB$$

By substitution the decomposed W in the equation(1) we will have

$$C = {I_t}^T \times AB \times T_t$$

$$S_{I_t}^T = {I_t}^T \times A$$

$$S_{T_t} = B \times T_t$$

$$C = S_{I_t}^T \times S_{T_t}$$

A is a mapping matrix which maps the image I_t into a semantic space and we represent the mapped image feature vector by S_{I_t} and B also is a mapping matrix which maps the text T_t into a corresponded semantic space and we represent the mapped text feature vector by S_{T_t}. Now both S_{I_t} and S_{T_t} are in the semantic space then the inner product of them will tell us about their similarity in this space. Now we have mapping matrix and we can easily map any given query image to the semantic space then using simple K-NN we can retrieve the images which are semantically similar to the given query image.

Non-Linear Model

As I elaborated in previous section, we can write the optimization problem in SVM fashion.

$$min\|\widetilde{W}\| \quad subject\ to: \ \forall ij\{{F^{ij}}^T \times \widetilde{W} \times M_{ij} \geq 1\}$$

The main drawbacks of this joint feature vector is this that it has very large dimension $m \times n$. In the case of linear SVM, I have to use this feature vector to find the corresponding weight vector. Also, another drawback is this that the semantic space can not be presented by linear models. However, in the case of non-linear SVM, we can simply concatenate an image feature vector I_i with the text feature vector T_j.

$$\tilde{F}^{ij} = \begin{bmatrix} I_i & T_j \end{bmatrix}$$

We can use $\tilde{F}$ instead of F because the space making by F is a subspace of the hyper space making by the non-linear kernel function on the $\tilde{F}$. Obviously, $\tilde{F}$ does not suffer of the drawback of high dimensionality. However, the main problem with the non-linear SVM is this that it will not give us a weight vector at the end. Therefore, we will not be able to create the mapping matrix to transform image vector from the visual feature space to the meaning space as I discussed in proposal. Because the the transformation is not any more linear so it can not be modelled by matrix multiplication. In the last part of this script I proposed a method to convert the visual feature space to the semantic feature space.

Semantic Image Descriptor

As I mentioned above using non-linear SVM we can not transform the visual feature space to the semantic feature space by any linear mapping functions. An interesting method would be such that taking advantage of texts' information to create a semantic space as follows:

1- Cluster the texts feature vectors into k clusters TC_i in which $i = 1, ..., k$.

2- For a given query image I extract its visual feature vector VI and create k joint feature vector(image-text) using the k text cluster centres $Joint_i = [VI, TC_i]$.

3- Apply the non-linear SVM (As discussed above) on the joint feature vectors $H(Joint_i)$. H is the SVM hypothesis and $H(Joint_i)$ is a real value that indicates the confidence of decision in SVM.

4- Save the k confidences H_i, $i = 1, ..., k$ as a k-dimensional vector to represent the image.

Experimental Results

All of my experiments have performed on the data set provided by Farhadi et al[1]. It contains 1000 images and for each images there are five sentences as descriptor. To set up the experiment, I needed to create feature vectors correspond to the texts and images.

To extract the text feature, first I tried to use the conventional bag-of-words model from the sentences in the dataset. However, because the number of sentences (variety of the vocabularies) are not enough to create an appropriate semantic space, I could not create a good descriptor. Therefore, I had to use some other precomputed text-analysed data. I used word similarity provided by NLP laboratory at UIUC. They have created a set of nouns, verbs and adjectives. For every arbitrary vocabulary item they provides a similarity measure to all of their nouns, verbs and adjectives. By concatenating these similarity measures we can make a feature vector for an arbitrary vocabulary item. In order to make a feature vector for a sentence one can easily compute the average of the feature vector corresponding to each vocabulary in the sentence. Another way would be to concatenate the feature vectors. I used the second way.

For the image descriptors I used two different image features. First, the visual feature used by Farhadi et al. Second, the Classemes feature by Torresani et al.

To reduce the dimensions of the text feature vector, I used a simple non-linear dimension reduction as follows:

1- Apply k-means clustering on the D-dimensional text feature vectors T_i. in which $k < D$. The cluster centres presented by Q_j, $j = 1, ..k$

2-For every text T_i, compute its $L2$ distance to all of the cluster centres Q_j and use these distances D_{ij} as a k-dimensional feature vector for that text.

To perform the learning, I considered variety of number of images and their assigned sentences as the training data (I chose it from 50 number up to 500). For the testing data, I got the remaining 500 images and their assigned sentences. I made a set of pairs of images and texts for both train and test. Then I create the joint feature vector (as I mentioned in the previous part) and labeled each pair by $+1$ when an image has same semantic as the text in the pair otherwise it labeled by -1.

To evaluate the result I made a plot in terms of number of training

example versus the accuracy achieving by SVM. As it is shown in Figure 1, I reported the both results of linear and non-linear SVM. As you can see for the linear SVM we do not gain so much more than chance which is %50. But for non-linear SVM we have fast increasing accuracy by increasing the training data. It means that the meaning space is not linear. Therefore, we can not work with linear models. The result of non-linear SVM is promising and we can hope that if we increase the number of training data we will achieve better accuracy. It is obvious that for this problem (I mean my project) we need much more data.

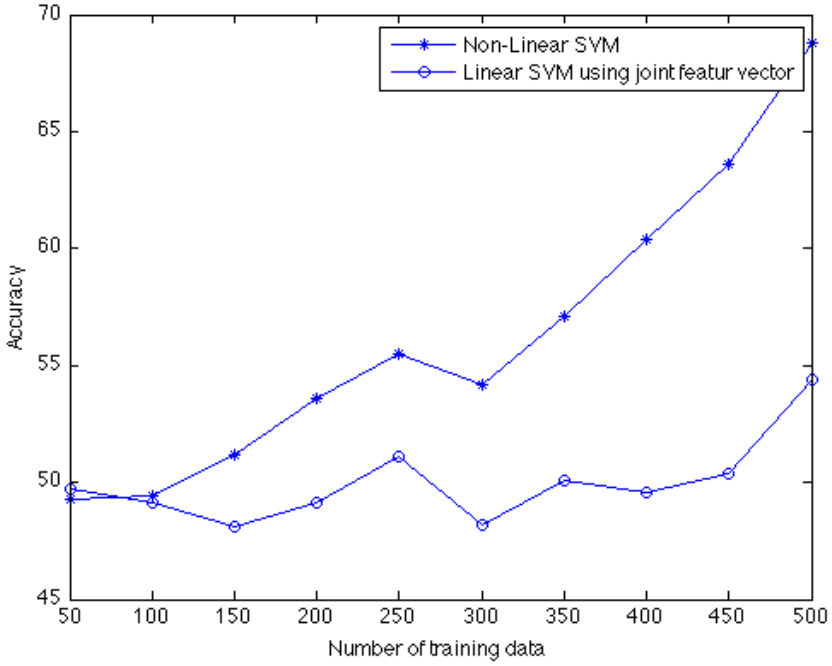

Figure 1: Quantitative evaluation

Figure 2 shows some qualitative result which I got randomly from the true-positives of the test data.

In figure 3 I showed the best result achieved by non-linear model using two different visual features. First, The visual feature introduced by Farhadi et al [1] and second, the classemes feature introduced by Torresani et al [4]. As you see in the plot there is not a very good accuracy by classmes. It is almost near to chance. The intuition behind that comes from the fact that

Figure 2: Quantitative evaluation

classemes are not designed to describe images that contain many objects. But it is very powerful when we have only one object in each image.

In figure 4 I tried show how much the semantic image descriptor can capture the semantic information of the images. I picked 5 random images (First column in the figure) and then for each of them I found their five nearest neighbour images by $L2$ distance of their semantic descriptor (Rows in the figure). As you see in each row we have some images that are semantically related to each other while their visual appearance are totally different.

In figure 5 I did an experiment which may seems not related to this project but the result was interesting. The semantic feature vectors came from a SVM. They are k dimensional real values (positive or negative). I simply bainarized them by thresholding them by zero. Then it become a k dimensional binary vectors. In figure 5 I used these binary vectors to do the same experiment as I did in figure 4. The result is still meaningful and we can see some semantically related image in each row.

In figure 6 I tried to numerically show that using the semantic descriptor instead of visual feature for images will not dramatically decrease the accuracy in the experiment that have shown in figure 3.

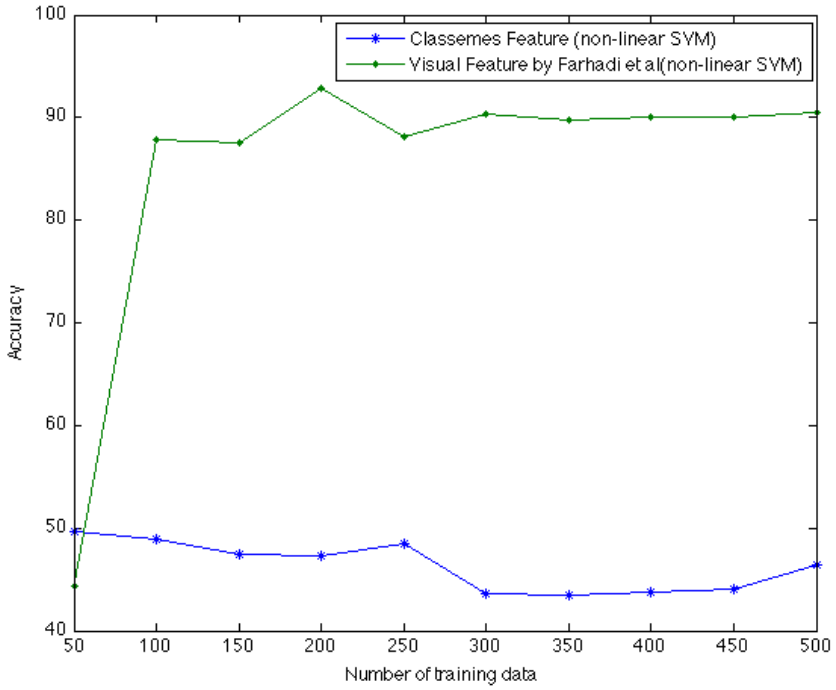

Figure 3: Best result using classemes and visual feature by Farhadi et al

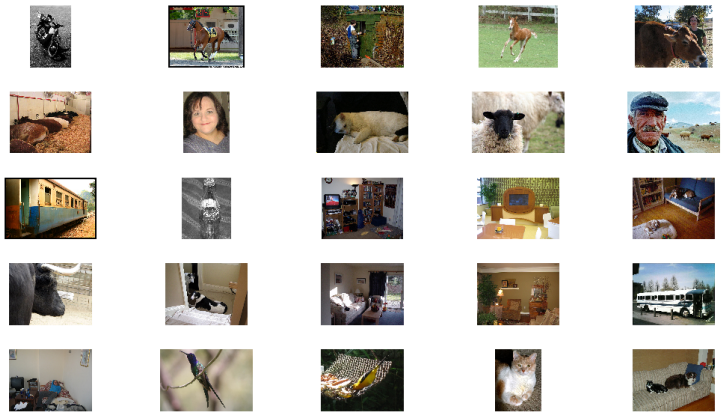

Figure 4: Nearest neighbours by semantic descriptor

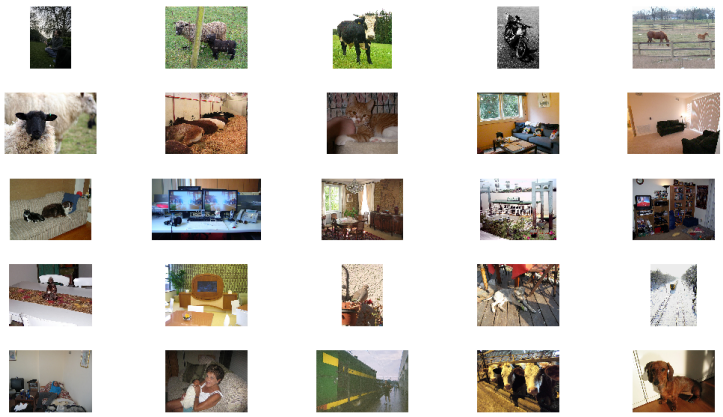

Figure 5: Best result using semantic descriptor and visual feature by Farhadi et al

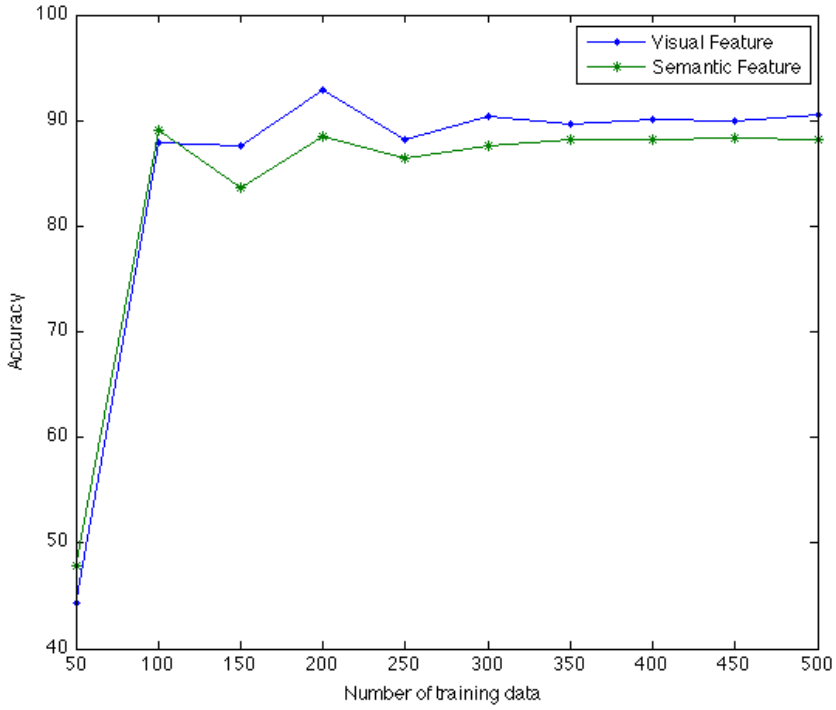

Figure 6: Nearest neighbours by binarized semantic descriptor

Bibliography

[1] Ali Farhadi, Seyyed Mohammad Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David A. Forsyth. Every picture tells a story: Generating sentences from images. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *ECCV (4)*, volume 6314 of *Lecture Notes in Computer Science*, pages 15–29. Springer, 2010.

[2] James Philbin and Andrew Zisserman. Near duplicate image detection: min-hash and tf-idf weighting.

[3] Lorenzo Torresani, Martin Szummer, and Andrew Fitzgibbon. Learning query-dependent prefilters for scalable image retrieval. In *CVPR*, pages 2615–2622. IEEE, 2009.

[4] Lorenzo Torresani, Martin Szummer, and Andrew Fitzgibbon. Efficient object category recognition using classemes. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *ECCV (1)*, volume 6311 of *Lecture Notes in Computer Science*, pages 776–789. Springer, 2010.