WEB AND PERSONAL IMAGE ANNOTATION BY MINING LABEL CORRELATION WITH RELAXED VISUAL GRAPH EMBEDDING

1PRASHANT SAKUNDE, 2ABHISHEK KAKALIJ, 3VISHWAJEET MANE, 4NIKHIL AKADE

1,2,3,4Department Of Computer Engineering, JSPM’s Rajarshi Shahu College of engineering, Tathawade, Pune
Email: Psakunde8@gmail.com, abhishekkakalis@gmail.com, Maneraj4@gmail.com, nikhilakade@gmail.com.

Abstract-Image processing refers to processing of a 2D picture by a computer. An image may be considered to contain sub images. Now a day the number of digital images rapidly increases. There are unlabeled images available, for that our system assigns automatic image annotation. We propose a system for image annotation by integrating label correlation mining and visual similarity mining into a join framework. We first construct a training image database which includes image visual features. A multilabel classifier is then train by simultaneously uncovering the shared structure common to different labels and the visual graph embedded label prediction matrix for image annotation. We apply the proposed framework to both image annotation and personal album labeling using the NUS-WIDE image dataset.

Keywords- Label Correlation Mining, Multilabel Learning, Personal Album Labeling, Web Image Annotation

I. INTRODUCTION

There are large amounts of digital images generated, shared, and accessed on different websites. The size of personal albums is getting larger. The growing number of personal images requires an effective retrieval and browsing mechanism in either a content or keyword based manner. Automatic image annotation enables conversion of image retrieval into text matching. Indexing and retrieval of text documents are faster and usually more accurate than that of raw multimedia data. Image annotation thus brings several benefits in image retrieval, such as high efficiency and accuracy. Usually, a single image may be associated with multiple labels, and the image annotation is a typical multilabel classification problem. A straightforward way to deal with this problem is to decompose it into several binary classification problems, with one for each label.

II. RELATED WORK

In this section we briefly discuss PCA (Principle Component Analysis). Principal Component Analysis is a way of identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences. Since patterns in data can be hard to find in data of high dimension, where the luxury of graphical representation is not available, PCA is a powerful tool for analyzing data. The other main advantage of PCA is that once you have found these patterns in the data, and you compress the data, i.e. by reducing the number of dimensions, without much loss of information. This technique used in image compression.

Step 1: Get some data Consider following example. In this example only two dimensions data is used, and the reason to chosen this is so that we can provide plots of the data to show what the PCA analysis is doing at each step.

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.7</td>
</tr>
<tr>
<td>2.3</td>
<td>2.7</td>
</tr>
<tr>
<td>2</td>
<td>1.6</td>
</tr>
<tr>
<td>1.5</td>
<td>1.6</td>
</tr>
<tr>
<td>1.1</td>
<td>0.9</td>
</tr>
<tr>
<td>1.5</td>
<td>1.6</td>
</tr>
<tr>
<td>1.1</td>
<td>0.9</td>
</tr>
</tbody>
</table>

Data = 

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.7</td>
</tr>
<tr>
<td>2.3</td>
<td>2.7</td>
</tr>
<tr>
<td>2</td>
<td>1.6</td>
</tr>
<tr>
<td>1.5</td>
<td>1.6</td>
</tr>
<tr>
<td>1.1</td>
<td>0.9</td>
</tr>
</tbody>
</table>

Data Adjust =

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.31</td>
<td>-1.21</td>
</tr>
<tr>
<td>0.39</td>
<td>0.99</td>
</tr>
<tr>
<td>0.99</td>
<td>2.9</td>
</tr>
<tr>
<td>1.29</td>
<td>1.09</td>
</tr>
<tr>
<td>0.49</td>
<td>0.79</td>
</tr>
<tr>
<td>0.19</td>
<td>-0.31</td>
</tr>
<tr>
<td>-0.81</td>
<td>-0.81</td>
</tr>
<tr>
<td>0.31</td>
<td>0.31</td>
</tr>
<tr>
<td>-0.71</td>
<td>-1.01</td>
</tr>
</tbody>
</table>

Step 2: Subtract the mean For PCA to work properly, we have to subtract the mean from each of the data dimensions. The mean subtracted is the average across each dimension. So, all the < values have _ subtracted, and all the = values have _ subtracted from them. This produces a data set whose mean is zero.

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.69</td>
<td>0.49</td>
</tr>
<tr>
<td>-1.31</td>
<td>-1.21</td>
</tr>
<tr>
<td>0.39</td>
<td>0.99</td>
</tr>
<tr>
<td>0.09</td>
<td>2.9</td>
</tr>
</tbody>
</table>

Figure: Plot of data

Step 2: Subtract the mean For PCA to work properly, we have to subtract the mean from each of the data dimensions. The mean subtracted is the average across each dimension. So, all the < values have _ subtracted, and all the = values have _ subtracted from them. This produces a data set whose mean is zero.
Step 3: Calculate the covariance matrix for an \( n \) -dimensional data set, we can calculate

\[
\frac{1}{(n-2)!2^n} \text{ different covariance values.}
\]

The definition for the covariance matrix for a set of data with \( n \) dimensions is:

\[
C_{n \times n} = (c_{i,j}, c_{i,j} = \text{cov}(\text{Dim}_i, \text{Dim}_j))
\]

Where \( C_{n \times n} \) a matrix with \( n \) rows and \( n \) columns, and \( \text{Dim}_x \) is the \( x \)th dimension.

Covariance=average of \((x \text{ mean} \ast y \text{ mean})\). The covariance matrix has 3 rows and 3 columns, and the values are this:

\[
C = \begin{pmatrix}
\text{cov}(x, x) & \text{cov}(x, y) & \text{cov}(x, z) \\
\text{cov}(y, x) & \text{cov}(y, y) & \text{cov}(y, z) \\
\text{cov}(z, x) & \text{cov}(z, y) & \text{cov}(z, z)
\end{pmatrix}
\]

For above example,

\[
\text{cov} = \begin{pmatrix}
.616555556 & .615444444 \\
.614444444 & .716555556
\end{pmatrix}
\]

Step 4: calculate the eigenvectors and eigenvalues of the covariance matrix. Covariance matrix is square; we can calculate the eigenvectors and eigenvalues for this matrix. Here are the eigenvectors and eigenvalues:

\[
eigenvalues = \begin{pmatrix}
.449083398 \\
1.28402771
\end{pmatrix}
\]

\[
eigenvectors = \begin{pmatrix}
-.735178656 & -.677873399 \\
.677873399 & -.735178656
\end{pmatrix}
\]

Step 5: Choosing components and forming a feature vector. Feature vector is just a fancy name for a matrix of vectors. This is constructed by taking the eigenvectors that we want to keep from the list of eigenvectors, and forming a matrix with these eigenvectors in the columns.

\[
\text{Feature Vector} = (e_{i1}, e_{i2}, e_{i3}, \ldots e_{in})
\]

For our example set of data, and the fact that we have 2 eigenvectors, we have two choices. We can either form a feature vector with both of the eigenvectors:

\[
\begin{pmatrix}
-.735178656 & -.677873399 \\
-.735178656 & .677873399
\end{pmatrix}
\]

Or, we can choose to leave out the smaller, less significant component and only have a single column:

\[
\begin{pmatrix}
-.677873399 \\
-.735178656
\end{pmatrix}
\]

Step 6: Deriving the new data set

This final step in PCA, and is also the easiest. Once we have chosen the components (eigenvectors) that we wish to keep in our data and formed a feature vector, we simply take the transpose of the vector and multiply it on the left of the original data set, transposed. Final Data = Row Feature Vector * Row Data Adjust Where Row Feature Vector is the matrix with the eigenvectors in the columns transposed so that the eigenvectors are now in the rows, with the most significant eigenvector at the top, and Row Data Adjust is the mean-adjusted data transposed i.e. the data items are in each column, with each row holding a separate dimension.

III. WORKING OF SYSTEM

First user gives set of unlabelled images. Then features of images are extracted. From these features graphs are generated such as weighted graph, Diagonal graph and Laplacian graph. Then PCA reduction is performed to get Eigen values and Eigen vectors. Then system performs prediction of label. Then system mine labels and select the nearest labels. Then System performs the AUC analysis and gives the labels to images.

IV. PROPOSED SYSTEM

To exploit label correlations for image annotation, it is reasonable to assume that different image labels are related and built on some underlying common structures. For example, different photos taken at the beach shared common characteristics, including sea, sky, and sand. We assume that there is a common subspace shared by multiple image labels. The final label of each image is predicted by its vector representation in the original feature space, together with the embedding in the shared subspace. Motivated by semi supervised learning, we construct a graph model...
CONCLUSION

In this paper, we have proposed a new framework for web and personal image annotation. We have proposed to simultaneously mine label correlation and visual similarities by integrating relaxed visual graph embedding into a joint framework. The label correlation has been learned by uncovering shared structure of different labels.

REFERENCES


★★★