MWDB Project
Phase – II
CSE 515 Fall- 18

Group 16

Aravamuthan Lakshminarayanan
1215338579
alaksh17@asu.edu

Chirag Bhansali
1215185491
cbhansal@asu.edu

Manthan Bhatt
1214918016
mbbhatt1@asu.edu

Parvika Singhal
1215146712
psingha4@asu.edu

Shubham Vipul Majmudar
1215200298
smajmuda@asu.edu

Yash Pande
1215159400
yspande@asu.edu

ABSTRACT

In the current era, the development in database management systems has inclined towards multimedia at a revolutionary rate. Along with the advancement of database technology that incorporates multimedia data, there are so many researches concentrating on the searching mechanisms in image databases for efficient storage, processing and retrieval. The competence and strength of a multimedia system are largely determined by the quality of the features available to it.
This project experiments with multimedia features and different techniques for data retrieval based on content and textual based descriptors, to refine the results by providing a set of images that are at the same time relevant to the query and to provide a diversified summary of it. Given Users, Images and Locations data from the Flickr database, it extracts 'k' related objects using different techniques like Latent Semantics Analysis, PCA, LDA etc.

KEYWORDS

SVD, PCA, LDA, Dimension reduction, Eigenvalue decomposition, Tensors

TERMINOLOGY

Color Histogram: The color histogram of an image defines the image color distribution by translating the color space and holding the count of pixels for each color zone.

HMMD Color Space: It regards the colors adjacent to a given color in the color space as the neighboring colors.
- *Hue* is the same as in the HSV space (0-360°)
- *Max* and *Min* are the maximum and minimum among the R, G, and B values
- *Diff* component is the difference between Max and Min

$$Sum = \frac{Max + Min}{2}$$

Minpair distance: This distance measure refers to sum over of the distance between the best pairs of instances under the given superclass and the superclass with which it is compared. For our case, as will be shown later, the superclasses are the locations that contain image instances (which in turn contain their respective feature vectors). The minpair distance of the given location with the i-th location is hence based on the minimum of the manhattan distances of each image vector of the given location with those of the images of i-th location.

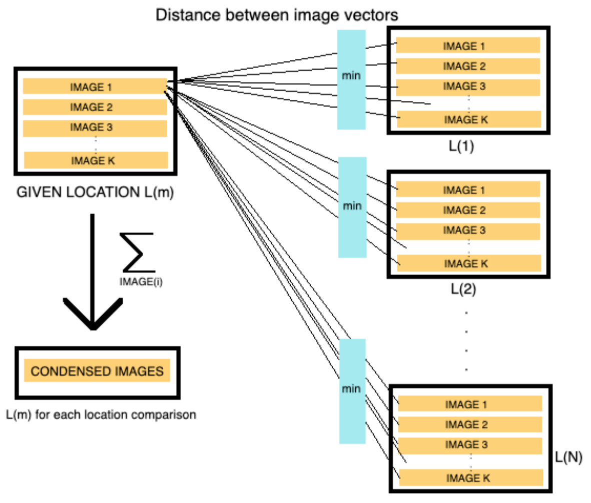


Figure 1 Minpair distance

It is to be noted here that this measure is asymmetric, and hence does not qualify as a metric.

Color space quantization: It renders a reference color set which is a representation of all color pixels in a color space and builds a lookup table which maps each pixel in the original color space to its belonged representative color bin in the reference color set. [1]

Color Naming Model: It is the action of assigning a linguistic color label to image pixels. The basic color terms (black, blue, brown, grey, green, orange, pink, purple, red, white, and yellow) were defined in the influential work on the color naming of Berlin and Kay. [4] Also called as Chip-based color naming.

Color Moments Model[15]: It indicates the statistical features (mean, standard deviation and skewness) of each Hue, Saturation and Value given the image is in HSV color-space.

Local Binary Pattern: This method was proposed by Ojala et al[11], and it involves finding the relation of a pixel with respect to its 8-connected immediate neighbors. A value of 1 is assigned if the intensity value of a neighbor is larger than or equal to that of the target pixel, and a value of 0 is assigned for intensities lesser than that of the target pixel. Now, the binary values are laid down right to left in an order that goes from the top left corner of the cell to the bottom right corner in a left-to-right, top-to-bottom fashion (Figure 4). This gives an 8-bit value for a pixel that is converted to decimal---this is the LBP value corresponding to this pixel. These values are calculated for all the pixels of the image and a histogram of 16 bins, each bin representing a range of these decimal values is obtained. This histogram is then normalized over the total number of pixels so as to generate a probability distribution over the bins.

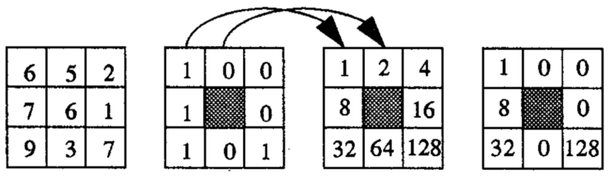


Figure 2 (left to right) Pixel neighborhood; 0-1 allocation; corresponding decimal value of the position of neighborhood; final decimal values for the positions, the sum of which gives the LBP value of the pixel[11]

Histogram of Oriented Gradients (HOG): The method of implementation of HOG[12] that has been used to generate the descriptor of the images is by calculating the histogram of oriented gradients split in 9 bins, on 3x3 image regions (resulting in 81 feature vectors). The oriented gradient of a pixel is can be given as :

$$gradient = arctan(\frac{\Delta y}{\Delta x})$$

$$magnitude = \sqrt{\Delta y^2 + \Delta x^2},$$

$where \Delta y and \Delta x refer to the difference of the pixel intensities located at [-1,0,1] and [-1,0,1]^T$

The bin closest to the gradient is identified and the value of the magnitude is stored.
Thus, when this window is moved across the image, each pixel location on the window now has a histogram of such gradients with the corresponding magnitudes.

Covariance Matrix: In probability theory and statistics, a covariance matrix (also known as dispersion matrix or variance-covariance matrix) is a matrix whose element in the i, j position is the covariance between the i^{th} and j^{th} elements of a random vector.

SVD: Singular Value Decomposition is a matrix decomposition method for reducing a matrix to its constituent parts to make certain subsequent matrix calculations simpler. Let $A \in \mathfrak{R}_r^{m \times n}[C_r^{m \times n}]$. Then there exist orthogonal (unitary) matrices $U \in \mathfrak{R}^{m \times m}[C^{m \times m}]$and $V \in \mathfrak{R}^{n \times n}[C^{n \times n}]$ such that

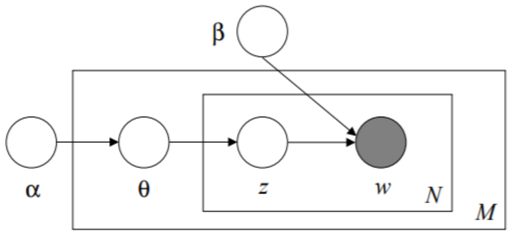
$A = U \Sigma V^T [U \Sigma V^H]$ where $\Sigma = \begin{pmatrix} S & 0 \\ 0 & 0 \end{pmatrix}$ and $S = diag(\sigma_1, \sigma_2, \sigma_3 \ldots \sigma_r)$with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$.[5] If A is a symmetric real $n \times n$ matrix, there is an orthogonal matrix V and a diagonal D such that $A = VDV^T$. Here the columns of V are eigenvectors for $A$and form an orthonormal basis for $\mathfrak{R}_n$; the diagonal entries of D are the eigenvalues of A.

PCA: Principal Component Analysis is also known as Karhunen-Loeve expansion [6], is a feature extraction and data representation technique widely used in the areas of pattern recognition and computer vision such as face recognition. It uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components.

Topic Modeling: It provides methods for automatically organizing, understanding, searching, and summarizing large data by discovering the hidden themes that pervade the collection, annotating the documents according to those themes and using these annotations to organize, summarize, and search the texts [7] .

LDA: Latent Dirichlet Allocation is a generative probabilistic model of a corpus that explains sets of observations by unobserved groups which in turn express why some parts of the data are similar. It assumes the following generative process for each document w in a corpus D:[8]

1. Choose $N \sim Poisson(\xi)$
2. Choose $\theta \sim Dirichlet(\alpha)$
3. For each of the N words w_n:
    a. Choose a topic $z_n \sim Multinomial(\theta)$
    b. Choose a word w_n from $p(w_n | z_n, \beta)$, a multinomial probability conditioned on the topic z_n.



{Known/Observed Variables : w, N, M}

Figure 3 LDA Plate diagram

Given the parameters α and β, the joint distribution of a topic mixture θ, a set of N topics z, and a set of N words w is given by :

$$p(\theta, \mathbf{z}, \mathbf{w} | \alpha, \beta) = p(\theta | \alpha) \prod_{n=1}^{N} p(z_n | \theta) p(w_n | z_n, \beta)$$

where $p(z_n | \theta)$ is simply θ_i for the unique i such that $z_n^i = 1$.Integrating over θ and summing over z, and taking the product of the marginal probabilities of single documents, we obtain the probability of a corpus:The probability of the corpus is given as below:

$$p(D | \alpha, \beta) = \prod_{d=1}^{M} \int p(\theta_d | \alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn} | \theta_d) p(w_{dn} | z_{dn}, \beta) \right) d\theta_d.$$

TF & DF: Term Frequency: It expresses frequent a term occurs in a document = count(word)/ total words; Document Frequency: The number of documents in a collection which contains a term t.

TF-IDF: Term Frequency-Inverse Document Frequency: It is a numerical statistic that reflects how important a term/feature is to a document in a collection of documents.

Tensors: It is a multidimensional array of numerical values. The dimensions of the tensor is the rank of the data matrix.

CP Decomposition: The key concept of the rank decomposition is to express a tensor as the sum of rank-one tensors. It is considered as generalization of the matrix SVD to tensor. A tensor x is decomposed as follows:

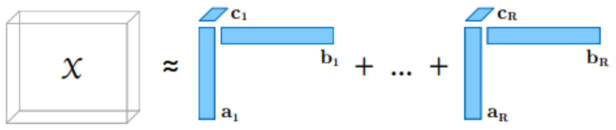


Figure 4 CP decomposition

Normalized Mean Square Error (NMSE) tolerance is given as-

$$\epsilon_k = \frac{\left\| \mathbf{Y}_{(n)} - \mathbf{A}_k^{(n)} \left(\bigodot_{m \neq n} \mathbf{A}_k^{(m)} \right)^T \right\|_F^2}{\left\| \mathbf{Y}_{(n)} \right\|_F^2},$$

PREPROCESSING

Since the frequencies in frequency-of-words representation are usually whole numbers, we round off the percentage/value to the nearest whole number after performing appropriate scaling/normalizing/processing for the respective model.

Color Naming Model (CN):

Latent aspect models are used extensively for the text analysis community as a tool to model documents as a mixture of several semantic –but a-priori unknown, and hence “latent”– topics. An image (document) is represented by a histogram indicating how many pixels are assigned to each bin (word). We used the image labels to define a prior distribution on the frequency of topics (color names) in documents. We normalized the distribution frequency of the topics by multiplying it with a factor of 100 as they were distrusted within the probabilistic range [0-1] (their summation is 1) and multiplying each with 100 helped us to scale them without losing the context.

For an image *d* with label l_d, the generative process applied as:

1) Sample θ_d (distribution over topics) from the Dirichlet prior with parameter α_{ld}

2) For each pixel in the image

a) sample *z* (topic, color name) from a multinomial with parameter θ_d

b) sample *w* (word, pixel bin) from a multinomial with parameter φ_d

Color Moments (CM):

The mean values for the Hue, Saturation and Value are being considered. As with other models, the preprocessing here scales these values appropriately to whole numbers.

K-Means Clustering: It is one of the methods that use the Euclidean distance, which minimizes the sum of the square Euclidean distance between the data points. It is used where there is high dimensionality, to find a low-dimensional representation of the documents to reduce computation complexity corresponding cluster centers.

Elbow Method: It is method of interpretation and validation of consistency within cluster analysis designed to help finding the appropriate number of clusters in a dataset.

Color Structure Descriptors (CSD):

Represents an image by its color distribution and its local spatial structure of the color, and its intended use is for still image retrieval. This additional information about color structure makes the descriptor sensitive to some image features to which the other color histogram is blind. [2] It is identical to a color histogram but not semantically. The color pixels of incoming images in any other color space is converted to the HMMD color space and re-quantized before extracting the structure histogram. [3]

The descriptor used 8x8 structuring element and defined using 64 bins color space quantization operating points. [10] The number of colors represented in a histogram is denoted by M; that is, the colors in the image are quantized into M different colors $c_0, c_1, c_2, \ldots, c_{M-1}$. The color structure histogram can be denoted by h(m), m=0, 1, …, M-1, where the value in each bin represents the number of structuring elements in the image containing one or more pixels with color "C_m". The final CSD is represented by 1D array of 8-bit quantized values, so that the range of the histogram is converted from [0, 1] values to 0,255. So, to find the LDA effectively we denormalized the quantization to convert it back in range of 0-255.

```
cv2.normalize(row, None, alpha=0, beta=255,
norm_type=cv2.NORM_MINMAX, dtype=cv2.CV_64F)
```

distance = (x-a)/(b-a) *(beta-alpha) + alpha (where a and b are lower, upper bounds of the input)

It scales and shift the array (histogram bin values) to a different range, ensuring norm of the input vector should be 0 and it uses Minkowski Distance Measure (L1, L2 or Infinity) where alpha is the minDistance and beta is the maxDistance. It normalizes the array by increasing the bit depth of the array.

Histogram of Oriented Gradients (HOG):

In order to perform LDA, a representation of the descriptor vector analogous to that of frequency-of-words can be created, in order to create a bag-of-words. This representation is done by taking one histogram at a time and dividing each bin value by the total weight of the histogram, and later converting it to a percentage value by multiplying it by 100.

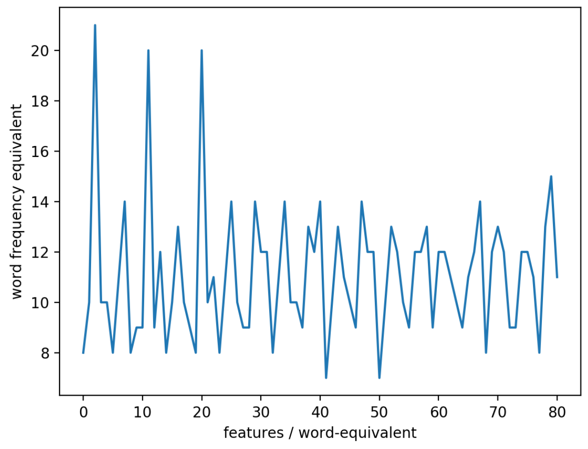


Figure 5 Pre-processed HOG feature vector for an instance of an image of the location Acropolis

Once the preprocessing has been done for all the 9 histograms, the data is now set to be given to the LDA function as input.

Local Binary Pattern (LBP):

To process the data of probability distribution (over the LBP value-bins) which is to be given as input to LDA, the probabilities are converted to whole number percentage values by taking the round of the values obtained after multiplying them each by 100.

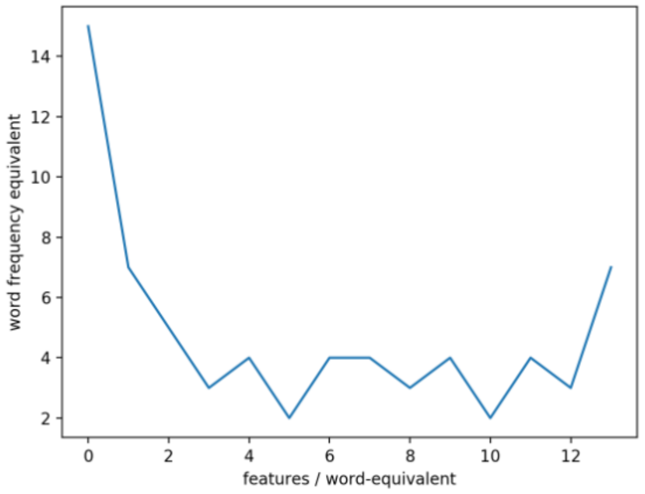


Figure 6 Pre-processed LBP feature vector for an instance of an image of the location Acropolis

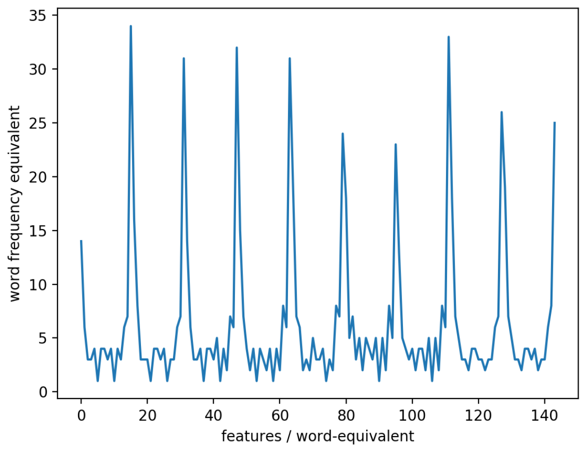


Figure 7 Pre-processed LBP3x3 feature vector for an instance of an image of the location Acropolis

For LBP3x3, LBP is calculated for a cell of size 3x3 and the result for each pixel location is concatenated, resulting in a feature vector of size 144 (16 bins x 9 pixel locations) for each image. An important thing to note is that this entire vector is a probability distribution, and not a set of probability distributions appended for each pixel location in the cell. Hence the same methodology as above is applied to convert the data matrix to frequency-of-words format.

Grey Level Run Length Matrix (GLRLM):
The grey level run length matrix (GLRLM) was introduced by Galloway to define various texture features. GLRLM assesses the run length of distribution of discretized grey levels in an image or in a stack of images. A run length is defined as the length of a consecutive sequence of pixels.

1	2	2	3
1	2	3	3
4	2	4	1
4	1	2	3

(a) Grey levels

		Run length j			
		1	2	3	4
i	1	4	0	0	0
	2	3	1	0	0
	3	2	1	0	0
	4	3	0	0	0

(b) $\mathbf{M_{m=\rightarrow}}$

		Run length j			
		1	2	3	4
i	1	4	0	0	0
	2	3	1	0	0
	3	2	1	0	0
	4	3	0	0	0

(c) $\mathbf{M_{m=\nearrow}}$

		Run length j			
		1	2	3	4
i	1	2	1	0	0
	2	2	0	1	0
	3	2	1	0	0
	4	1	1	0	0

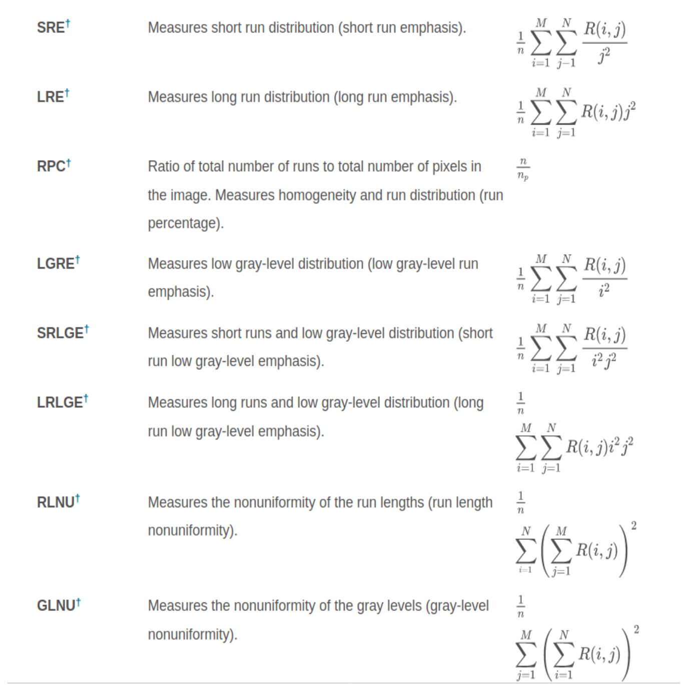
(d) $\mathbf{M_{m=\uparrow}}$

		Run length j			
		1	2	3	4
i	1	4	0	0	0
	2	3	1	0	0
	3	4	0	0	0
	4	3	0	0	0

(e) $\mathbf{M_{m=\nwarrow}}$

Figure 8 Grey Level run length matrices for the (a) 0° (b) 45° (c) 90° and (d) 135° directions. In vector notation these directions are m = (1,0), m = (1,1), m = (0,1), m = (-1,1), respectively

with the same grey level along direction.[14] Our dataset in GLRLM is measured across 4 directions and has 11 statistical measures in each direction namely, Short Run Emphasis (SRE), Long Run Emphasis (LRE), Gray-Level Non-uniformity (GLN), Run Length Non-uniformity (RLN), Run Percentage (RP), Low Gray-Level Run Emphasis (LGRE), High Gray-Level Run Emphasis (HGRE), Short Run Low Gray-Level Emphasis (SRLGE), Short Run High Gray-Level Emphasis (SRHGE), Long Run Low Gray-Level Emphasis (LRLGE), Long Run High Gray-Level Emphasis (LRHGE).



SRE†	Measures short run distribution (short run emphasis).	$\frac{1}{n}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{R(i,j)}{j^2}$
LRE†	Measures long run distribution (long run emphasis).	$\frac{1}{n}\sum_{i=1}^{M}\sum_{j=1}^{N}R(i,j)j^2$
RPC†	Ratio of total number of runs to total number of pixels in the image. Measures homogeneity and run distribution (run percentage).	$\frac{n}{n_p}$
LGRE†	Measures low gray-level distribution (low gray-level run emphasis).	$\frac{1}{n}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{R(i,j)}{i^2}$
SRLGE†	Measures short runs and low gray-level distribution (short run low gray-level emphasis).	$\frac{1}{n}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{R(i,j)}{i^2 j^2}$
LRLGE†	Measures long runs and low gray-level distribution (long run low gray-level emphasis).	$\frac{1}{n}\sum_{i=1}^{M}\sum_{j=1}^{N}R(i,j)i^2 j^2$
RLNU†	Measures the nonuniformity of the run lengths (run length nonuniformity).	$\frac{1}{n}\sum_{i=1}^{N}\left(\sum_{j=1}^{M}R(i,j)\right)^2$
GLNU†	Measures the nonuniformity of the gray levels (gray-level nonuniformity).	$\frac{1}{n}\sum_{j=1}^{M}\left(\sum_{i=1}^{N}R(i,j)\right)^2$

Figure 9: Selected Statistical measures. Where *R(I,j)* is an element of the RLM, n is the total number of runs, n_p is the number of pixels in the image, N is the longest run, and M is the number of gray levels.

Each statistical measure gives different importance to either run length or grey intensity or both equally. We preprocess the data by dividing the maximum value from all statistical measures in all directions one-by-one to find the tangible range and then multiply it by 100 for the entire data.

TASK 1

Goal Description

To implement a program which lets the user choose among

1. user-term vector space
2. image-term vector space, or
3. location term vector space,

and then, given, a positive integer value, k, identifies and reports the top-k latent semantics/topics in the corresponding term space using

1. PCA,
2. SVD, or
3. LDA.

Each latent semantic should be presented in the form of term-weight pairs, ordered in decreasing order of weights.

Assumptions

The text file names does not change and content is space separated. The libraries methods implementation doesn't change impacting our use-case.

Implementation

1. We first read the textual descriptor files for each document type and created a matrix for the unique terms with their TF-IDF or TF values vs document id. (*We chose TF-IDF as the textual descriptor for SVD and PCA, because it reflects how important a term is to a document in a collection or corpus.We chose TF for LDA as LDA is based on term counts.)*
2. For storing the data in-memory or caching we are using Redis.
   Redis Internals: An in-memory key-value pair NoSQL data store often used for web application sessions, transient data and as a broker for task queues. Redis-py uses a connection pool to manage connections to a Redis server. By default, each Redis instance created will in turn create its own connection pool.
3. We then used skLearn libraries to calculate the PCA, SVD and LDA by passing the created matrix to those libraries methods.
   (We initially tried to write our own implementation but it wasn't efficient enough. We tested other inbuilt libraries like NumPy, Gensim, Surprise but Sklearn Libraries outperformed.)
   sklearn.decomposition **import** TruncatedSVD-
   This transformer performs linear dimensionality reduction by means of truncated singular value decomposition (SVD), that only computes the k largest singular values, where k is a user-specified parameter..Truncated SVD works on term count/tf-idf matrices as returned by the vectorizers in sklearn.feature_extraction.text. It uses a fast randomized SVD solver. Mathematically, truncated SVD applied to training samples X produces a low-rank approximation X: $X \approx X_k = U_k \Sigma_k V_k^T$. Computing a low-rank approximation consists of constructing a low-dimensional subspace that captures the action of the matrix and restricting the matrix to the subspace and then computing a standard factorization (QR, SVD, etc.) of the reduced matrix.[9] After this operation, $U_k \Sigma_k^T$ is the transformed training set with k features (called n_components in the API). To also transform a test set X, we multiply it with $V_k : X' = X V^k$.
   PCA **-** Linear dimensionality reduction using Singular Value Decomposition of the data to project it to a lower dimensional space.It uses the LAPACK implementation of the full SVD or a randomized trun-

cated SVD by the method of Halko et al. 2009[9], depending on the shape of the input data and the number of components to extract.
LDA - Implements online variational Bayes algorithm and supports both online and batch update method. While batch method updates variational variables after each full pass through the data, online method updates variational variables from mini-batch data points. It is applied on a "document-term" matrix, to decompose it into a "topic-term" matrix and a "document-topic" matrix. While "topic-term" matrix is stored as *components_* in the model, "document-topic" matrix can be calculated from *transform* method. The resultant term-weighted matrix is product of right-factor matrix with the core matrix for SVD and PCA. The k-latent semantics would be the right factor matrix.

TASK 2

Goal Description

To implement a program that, after the top-k latent semantics are identified, given

1. a user ID,
2. an image ID, or
3. a location ID,

the system also identifies the most related 5
– user IDs,
– image IDs, and
– location IDs
using the above-calculated k latent semantics and list the matching scores.

Assumptions

While finding objects (like images and users) closest to one location, we get approximately 300 images and 25 users being related to one location. So for comparing all 300 images to the entire image dataset using cosine similarity we do around 300*10000*30000 i.e. 900 million multiplications which is computationally inefficient. Thus, we average out all the 300 objects to 1 and then compare it to 10000 objects in the reduced latent semantics after PCA/SVD or LDA and find the closest 5 after sorting. We assume that this approach is not impedently lossy.

Implementation

1. In this task we have to compare each object (user, image and location) with every other object, so we use xml files to create 6 hashmaps of interrelated objects namely, location_to_user, user_to_location,image_to_location,location_to_image, image_to_user and user_to_image.
2. We then use the given object (user, image or location) id as key and get the other related objects to the id which are not of the type of this object. For example: using the user id we can get multiple image ids and location ids related to that user from the xml.
3. After getting the image ids we compare 1 object which is average out of all the related images for the user input, to all the images in the dataset and find the closest image in the same feature space using the cosine similarity. The images closest to this average are the ones closest to our user object.
4. The feature space is the reduced feature space of latent semantics for all the objects obtained by SVD, PCA or LDA as specified by user in task1. We have used redis for retrieving id to row number and vice-a-versa for all the objects which was created in task1. *All the maps are stored in-memory using redis and dictionaries for faster retrieval. Thus accessing object row number from object id and closest id using the row number has order of O(1).*

TASK 3

Goal Description

To implement a program which, given a visual descriptor model (CM, CM3x3, CN, CN3x3, CSD, GLRLM, GLRLM3x3, HOG, LBP, LBP3x3) and value "k", – first identifies (and lists) k latent semantics using

1. PCA,
2. SVD, or
3. LDA,

on the corresponding image feature space, then given an image ID, identifies the most related 5

❏ image IDs and
❏ location IDs

using these k latent semantics and list the matching scores.

Assumptions

The representation of visual descriptor values are appropriate to LDA dimensionality reduction

Implementation

1. In order to accomplish this task we first use the value of 'k' to do the dimensionality reduction and find k-latent semantics using either of the three choices we have PCA, SVD and LDA of then given visual descriptor''s vector space.

2. Once the reduced image_visual vector space is found we get 'k' feature array values of the inputId mentioned and use the same to get similarity with other images and sort the compared images with their decreasing order of similarity scores.
   To obtain the similarity scores we have used L2 similarity since it gives us one best measure of comparison between entities in visual descriptor vector space. L2 distance between two vectors p and q in n dimensions is defined as below:

$$d(\mathbf{p},\mathbf{q}) = d(\mathbf{q},\mathbf{p}) = \sqrt{(q_1-p_1)^2 + (q_2-p_2)^2 + \cdots + (q_n-p_n)^2}$$
$$= \sqrt{\sum_{i=1}^{n}(q_i-p_i)^2}.$$

3. Once we have the similarity scores of comparison in descending of the scores we mark the first 5 images Id with similarity score as the output for 5 most similar images compared to the imageId, and for finding the top 5 locations we use list of the compared iterate over it till we get 5 distinct locations

TASK 4

Goal Description

To implement a program which, given a location ID, a visual descriptor model (CM, CM3x3, CN, CN3x3, CSD, GLRLM, GLRLM3x3, HOG, LBP, LBP3x3), and value k,
– first identifies (and lists) k latent semantics corresponding to the given location using
1. PCA,
2. SVD, or
3. LDA

and then identifies and lists the most related 5 locations using these k latent semantics and list the matching scores.

Assumptions

Minpair distance is an accurate measure of the distance between two locations.

Implementation

1. The user gives the location id, visual descriptor model, and the number of latent semantics as input. The id is used to obtain the corresponding location name from the '.xml' file.
2. The top 'k' latent semantics of all the images for the given model and location are identified.
3. The minpair distance of all the locations with respect to the given location is calculated using these 'k' features of the new image vectors.
4. Locations are arranged in ascending order of distances from the given location and top 5 locations are returned.

TASK 5

Goal Description

Implement a program which, given a location ID and value k
– first identifies (and lists) k latent semantics corresponding to the given location using
1. PCA,
2. SVD, or
3. LDA

applied across all visual descriptor models (as before, the choice of the dimensionality reduction algorithm to be used is left to the user), and then
– identifies and lists the most related 5 locations using these k latent semantics (list also the matching scores).

Assumptions

Each visual descriptor model has an equal say while calculating the location similarity (distance).
Minpair distance is an accurate measure of the distance between two locations.

Implementation

1. The user gives the location id and the number of latent semantics as input. The id is used to obtain the corresponding location name from the '.xml' file.
2. The top 'k' latent semantics of all the images for all the visual descriptor models for the given location are identified and stored.
3. The minpair distance of all the locations with respect to the given location is calculated using these 'k' features of the new image vectors, for all the models.
4. Locations are arranged in ascending order of distances from the given location for each visual descriptor model, resulting in 10 such 'ranklists'.
5. The average position of every location over all the ranklists is calculated---this gives us the overall ranklist.
6. Top 5 locations from the overall ranklist are returned.

TASK 6

Goal Description

Implement a program which, given a value k,
– creates a location-location similarity matrix,

– performs SVD on this location-location similarity matrix,
– reports the top-k latent semantics.
Each latent semantic should be presented in the form of location (name)-weight pairs, ordered in decreasing order of weights.

Assumptions

It is assumed that textual descriptors are sufficient to find the similarity between locations

Implementation

1. We first import the task 1 file from the '*api'* folder and math file from the '*lib'* folder.
2. The user is asked to enter the value of *k*, the number of latent semantics required.
3. We create an instance of Task 1 since in this task we created the function to generate the 3 object-feature matrices.
4. We then load the matrix of Location in *'location_arr'*.
5. We find the matrix multiplication of *'location_arr'* and its transpose (This gives us the location-location similarity based on the textual descriptor, we store this matrix in *'sim_matrix_td'*.
6. After that, we find the SVD of the resultant similarity matrix, to get to the diagonal matrix of the same.
7. Lastly, we print the location (name)-weight pairs.

TASK 7

Goal Description

Implement a program which, given a value k,
– creates an user-image-location tensor (based on number of terms shared by all three),
– performs rank-k CP decomposition of this tensor, and
– creates k non-overlapping groups of users, images, and locations based on the discovered latent semantics.

Assumptions

We have set the tolerance for the CP decomposition at 10e-2, so that the number of iterations are less.

Implementation

1. User is asked to enter the value of k.
2. Arrays for user,image and locations are created each having the terms corresponding to the individual ids.
3. For filling the 3d array for the user-image-location we find the number of common terms in all of them.
4. Finally we use the 'parafac' method for calculating the factor matrices with parameters as the 3d array,rank as k and tolerance as 10e-2.
5. After getting the factors matrices U1, U2 and U3 we calculate the best value of k for K-means clustering using the elbow graph. We set the values of k for U1, U2 and U3 when the graph converges to the least change in slopes by setting the threshold of change in slope to be less than 0.01.
6. The k means clustering objects when transformed gives us non-overlapping users, locations and images from CP decomposed U1, U2 and U3.
7. CP decomposition gives soft clustering, while k-means with an appropriate value of 'k' gives non-overlapping sets of users, images and locations in newly discovered latent semantics.

SYSTEM AND INTERFACE REQUIREMENTS

1. Python 3 or above
2. OS – No Dependency (This code has been run and tested on Windows and Mac)
3. 64-bit Machine
4. RAM – At least 8GB
5. Python Libraries: math, xml.etree.ElementTree, sklearn.LDA, sklearn.PCA, sklearn.TruncatedSVD, pandas, Redis, numPy

EXECUTION INSTRUCTIONS

1. Open a terminal and browse into Project directory - MWDB-P2.
   Data set should be in the current directory.
2. Install Redis.
3. Run Setup.py
4. Run Main.py

CONCLUSIONS

The data matrices obtained on the new features display the most variance in the initial dimensions, and the variance tapers off progressively as we move towards the lower features. Thus most of the information of the data is stored in only a fraction of the original dimensions.
Each of the tasks are implemented according to the methods described in the implementation section and the outputs for each of the tasks are generated and attached in the folder Sample Outputs.

BILIOGRAPHY

[1] Wang, Jia, Wen-jann Yang, and Raj Acharya. "Color space quantization for color-content-based query systems." Multimedia Tools and Applications 13.1 (2001): 73-91.
[2] Abdelali, Abdessalem Ben, et al. "A study of the color-structure descriptor for shot boundary detection." International Journal of Sciences and Techniques of Automatic control and computer engineering (2009): 956-971.
[3] Mane, Pranoti, and Dr NG Bawane. "Comparative performance evaluation of edge histogram descriptors and color structure descriptors in content based image retrieval." IJCA Proceedings on NCIPET 6 (2013): 5-9.
[4] B. Berlin and P. Kay, Basic color terms: their universality and evolution, Berkeley: University of California, 1969
[5] Klema, Virginia, and Alan Laub. "The singular value decomposition: Its computation and some applications." IEEE Transactions on automatic control 25.2 (1980): 164-176.
[6]Abdi, Hervé, and Lynne J. Williams. "Principal component analysis." *Wiley interdisciplinary reviews: computational statistics* 2.4 (2010): 433-459.
[7] Blei, David M. "Probabilistic topic models." *Communications of the ACM* 55.4 (2012): 77-84.
[8] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." *Journal of machine Learning research* 3.Jan (2003): 993-1022.
[9] Halko, Nathan, Per-Gunnar Martinsson, and Joel A. Tropp. "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions." *SIAM review* 53.2 (2011): 217-288.
[10] Manjunath, Bangalore S., et al. "Color and texture descriptors." IEEE Transactions on circuits and systems for video technology 11.6 (2001): 703-715.
[11] Ojala, T., Pietikäinen, M., Harwood, D. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. IAPR International Conference on Pattern Recognition, vol. 1, 1994, 582 - 585.
[12] Ludwig, O., Delgado, D., Goncalves, V., Nunes, U. Trainable Classifier-Fusion Schemes: An Application To Pedestrian Detection. Conference On Intelligent Transportation Systems, 2009.
[13] Rabanser, Stephan, Oleksandr Shchur, and Stephan Günnemann. "Introduction to Tensor Decompositions and their Applications in Machine Learning." arXiv preprint arXiv: 1711.10781 (2017).
[14] Zwanenburg, A., Leger, S., Vallières, M. and Löck, S., 2016. Image biomarker standardisation initiative. arXiv preprint arXiv:1612.07003.
[15] Stricker, M., Orengo, M. Similarity of color images. SPIE Conference on Storage and Retrieval for Image and Video Databases III, vol. 2420, 1995, 381 - 392.
[16] Shi, Ming, Dan Li, and Jian Qiu Zhang. "An Alternating Bayesian Approach to PARAFAC Decomposition of Tensors." *IEEE Access* 6 (2018): 36487-36499.