CSE 515 Multimedia and Web Databases

October 29, 2024

Phase #3

(Due December 1st 2024, midnight)

Description: In this project, you will experiment with

- video features,
- vector models,
- indexing,
- classification,
- relevance feedback.

Important notes:

- Each group will select 3 latent models, differing in the underlying feature space and dimensionality reduction technique. Tasks 1 through 4 will be implemented considering each of these latent models.
- If your group has 3 members instead of 4, please skip task 4.
- If your group has 2 members instead of 4, please skip tasks 3 and 4.

Project Tasks:

- **Task 0a:** Implement a program which computes and prints the "inherent dimensionality" associated with each unique label in *target_videos*.
- **Task 1:**
  - **Task 1a:** Implement a program which, for each unique label l in *target_videos*, computes the corresponding c most significant clusters associated with the even numbered *target_videos* using spectral clustering (you cannot use a library for this purpose); the resulting clusters of videos should be visualized both

    - as differently colored point clouds in a 2-dimensional MDS space (you can use a library for MDS), and
    - as groups of video thumbnails
  - **Task 1b:** Implement a program which computes the c most significant label clusters for the *target_videos* using k-means clustering (you cannot use a library for this purpose); the resulting clusters of labels should be visualized both
    - as differently colored point clouds in a 2-dimensional MDS space (you can use a library for MDS), and
    - as groups of labels.

- **Task 2:** Implement a program which,
  - for a given videoID, v_i, of an odd numbered *target_video* or a *non_target_video*
  - and a value m

  predicts the most likely m labels using each of the following classifier models:

  - **Task 2a:** k-NN classifer for a user specified k (you cannot use a library for this purpose),
  - **Task 2b:** SVM clasifier (you cannot use a library for this purpose)

  When the videoID, v_i, is of an odd numbered *target_video*, the system should also output per-classifier *accuracy* value, defined as

  $$accuracy\_classifier(v_i, m) = \frac{m - rank_i + 1}{m}$$

  where $rank_i$ denotes the rank of the true label of video v_i among the returned m labels. If the true label has not been returned, than

  $$accuracy\_classifier(v_i, m) = 0$$

- **Task 3:**
  - **Task 3a:** Implement a Locality Sensitive Hashing (LSH) tool (for Euclidean distance) which takes as input (a) the number of layers, L, (b) the number of hashes per layer, h, and (c) a set of vectors as input and creates an in-memory index structure containing the given set of vectors. See

    > ”Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions” (by Alexandr Andoni and Piotr Indyk). Communications of the ACM, vol. 51, no. 1, 2008, pp. 117-122.

  - **Task 3b:** Implement a similar video search tool using this index structure storing all target and non-target videos for a visual model of your choice (the visual model must have at least 256 dimensions): for a given query videoID, v_i, and integer t, the tool
    - presents, in decreasing order of similarity, the thumbnails of the most similar t videos,
    - outputs the numbers of unique and overall video candidates considered during the process.

- **Task 4:** Let us consider the tag set "Liked (should have had a higher ranking)" and "Disliked (should have had a lower ranking)". Implement
  - **Task 4a:** a decision-tree based relevance feedback system (you cannot use a library for this purpose),
  - **Task 4b:** a k-NN based relevance feedback system (you cannot use a library for this purpose),

  which enable the user to tag some of the results returned by Task 3b and then obtain a new set of ranked results, relying on the feedback system selected by the user.

Deliverables:

- Your code (properly commented) and a README file.
- Your outputs for the provided sample inputs.
- A short report describing your work and the results.

Please place your code in a directory titled "Code", the outputs to a directory called "Outputs", and your report in a directory called "Report"; zip or tar all off them together and submit it through the digital dropbox.