Bag-of-Words Video Clustering

Elijah Houle

Abstract

The bag-of-words model has been widely used in the domain of text classification and clustering. Similarly, the computer vision community has had success implementing the model for human action and scene recognition. However, researchers have made few, if any, attempts to consider the domain of online video sharing, in which video clips often contain content derived from others. This project extends the bag-of-words model to clustering modified videos with equivalent content.

Research Problem

Many approaches for representing a video as a bag of words exist in the context of human action and scene recognition. However, the bag-of-words approach may also prove useful within a system where many videos consist of clips derived from others, such as YouTube and similar sharing services. This project focuses on applying the bag-of-words approach to video for clustering similar video sequences.

In this context, two videos are "similar" if they contain similar images (itself being a component of the research question). Temporal information may be safely discarded for this purpose, because a modification may consist of reordering the keyframes, in which case the derived video should still be clustered with the original. This distinction simplifies the problem and makes it suitable for bag-of-words approaches.

The research question has the following components:

1. What represents video words? Is a frame a word or a bag of words itself?
   - Keyframe extraction for selecting relevant images
     - The use of keyframes is necessary to cut down on space. Because consecutive frames are redundant, keyframe selection agglomerates a small series of frames into the most representative image.
   - Feature extraction and encoding of words
     - Histogram, SIFT descriptor, etc.
2. Which similarity metric is used?
   - Depends on how frames are encoded as words
     - Euclidean distance, cosine similarity, etc.
3. Which type of clustering should be used?
   - Compare different types of clustering
     - Agglomerative, k-means, SVM
     - Distance (single-linkage, complete-linkage, etc.)

The first part, feature extraction and encoding the video as a bag of words, makes up the bulk of the initial research. Once a video is represented by a bag of words, various algorithms for standard bag-of-words clustering can be applied to the domain and compared through experimentation.

Related Works

Many papers apply the bag-of-words model to human action or scene recognition, but these features might not apply to general videos. Thus, more generic feature extraction is desired.

Jiang, Ngo, and Yang (2007) surveyed various keypoint detectors and SVM kernel functions along with their impacts on bag-of-features performance. In this model, each image is a bag of keypoint clusters. This is extended to video by using one keyframe per shot as the query image. However, the model could extend to video differently, by compressing each image's histogram into its own feature.

In this vein, Zhou et al. (2008) used frames as features by representing video as a bag of SIFT feature vectors. Depending on the succinctness and robustness of SIFT, these vectors or analogues may be used as "words" for this domain.

Cheung and Zakhor (2005) introduced a fast, robust signature method called ViSig for video similarity measurement. While this clustering does not use bag-of-words, it can be used as a benchmark for comparison.

Infrastructure

The main infrastructure is written in Python, using FFMS for reading from seed videos and OpenCV for generating temporary videos from the seeds. The package scikit-learn, built on NumPy, SciPy, and matplotlib, is used for clustering.

The seed videos were the top 200 trending videos on YouTube, downloaded using youtube-dl. During each experiment, a number of videos are generated by mixing frames from two seeds using parameters:

- P = number of frames to grab from primary video (class label)
- S = number of frames to grab from random secondary video (noise)
- Maximum number of videos to generate from seed (limit for random size of class)
- Filters/transformations to apply (variable)

Thus, a unique dataset for each experiment is generated, containing an random number of derived videos for each original sequence, each labelled identically. In other words, a "class" of videos consists purely of an original sequence and all sequences derived from it.

Feature Extraction

The two methods used for encoding frames as words are adapted from basic perceptual hashing algorithms. Each keyframe (I-frame) is "hashed" to a bitstring, with the length depending on the number of pixels taken into account. First, the frame is resampled to MxN pixels and converted to grayscale. Then, for each pixel, *aHash* (average hashing) sets the corresponding bit to 1 if its value is greater than or equal to the mean, while *dHash* (difference hashing) does so if its value is greater than or equal to the previous. Otherwise, the bit is set to 0. This implementation flattens the image so that rows wrap around in *dHash*, but early testing did not demonstrate a clear difference either way.

To withstand transformations such as cropping, later experiments hash $(M/2)$x$(N/2)$ quarter blocks of the frame in addition to the whole frame (see Results). In order to compensate for common features, tf-idf vectorization is performed on the resulting bags before clustering.

Clustering

The two clustering algorithms used are scikit-learn's implementations of k-means and Ward hierarchical clustering. Most of the default parameters are kept; however, n_init is set to 1 rather than 10, meaning that only 1 centroid seed is used rather than running multiple trials with different seeds. $n\_clusters$ is set to the true number of classes. Future work will examine the effect of multiple k-means runs ($n\_init > 1$) as well as the use of heuristics in estimating the number of clusters.

Metric

The main goal of this research is to cluster video sequences with others that derive content from the same source. To test encoding and clustering methods, each item in a dataset is labelled with the identifier of the original clip. The best method would be one in which video clusters accurately match their true classes.

Adjusted mutual information (AMI) is used to measure the agreement between cluster assignments and true class labels. AMI is adjusted against chance, returning a number $[-1.0, 1.0]$, where 0.0 is equivalent to a random (chance) assignment and 1.0 is a perfect labeling. Therefore, AMI $<$ 0.0 implies that the assignment is worse than chance and AMI $>$ 0.0 is better than chance.

Results and Discussion

After initial setup, the first clustering using the entire dataset serves to validate the model with scaling to $(width = 640, height = 480)$ as the only transformation. Although this may result in distortion for some videos with vastly different aspect ratios (which may explain why the results are lower than expected), assignment for all schemes is significantly better than chance. Each of the tables in this section shows the AMI for each scheme.

In Table 1, the seed videos are excluded from the dataset (videos are expected to cluster with others derived from the same seed). Frames are selected at random.

	aHash	dHash
k-means	0.533144269036	0.458820993492
Ward	0.584658068632	0.527118369274

Table 1: Word as hash of entire frame. Frames selected at random; seeds excluded from dataset. Scaling only. 410 videos generated with 200 clusters. $(P, S) = (100, 10), (M, N) = (4, 4)$.

Now that the model is validated with the above assumptions, dataset generation can begin to approximate real-world conditions. Because selecting individual frames at random is an unnatural way to include video content from another source, all of the following experiments select a contiguous sequence of frames from the seeds. In addition, seeds are included in the dataset so that all videos have at least one match. While words previously consisted of the entire frame's hash, they now consist of each $4x4$ quarter block hash (top-left, top-right, bottom-left, bottom-right) in order to withstand later transformation. Table 2 shows an overall decrease in AMI. This is due to fewer keyframes (more consecutive frames from same shot) and hashes being smaller, which lowers discriminative power and thus increases positive rate.

	aHash	dHash
k-means	0.449813755011	0.415802697766
Ward	0.483065659146	0.39643336826

Table 2: Words as hashes of quarter-frames. Frames selected as contiguous sequences; seeds included in dataset. Scaling only. 275 videos generated with 50 clusters. $(P, S) = (400, 10), (M, N) = (8, 8)$.

Including the entire frame's hash back with the quarter hashes yields better results, as in Tables 3 and 4. This illuminates an interesting insight: the bag-of-words model works best for video when frames can be represented using multiple words. Otherwise, the chance of matching decreases because of slight differences in features.

In the next experiments, a uniform transformation is applied to each frame of the generated videos with the seed videos remaining untouched. The first transformation is cropping 50 pixels off each side, which demonstrates the necessity of hashing subimages as well as the whole frame. Table 5 shows the result of cropping with hashing as before,

	aHash	dHash
k-means	0.579116928501	0.485880328779
Ward	0.568734121065	0.565466884595

Table 3: Words as hashes of quarter-frames and entire frame. Scaling only. 272 videos generated with 50 clusters. $(P, S) = (400, 10), (M, N) = (8, 8)$.

	aHash	dHash
k-means	0.347006637323	0.294915974058
Ward	0.534182656989	0.470320338044

Table 4: Words as hashes of quarter-frames and entire frame. Scaling only. 596 videos generated with 200 clusters. $(P, S) = (400, 10), (M, N) = (8, 8)$.

while Table 6 shows it without quarter-frame hashes (just the hash of the whole frame). Notice the drop in AMI.

	aHash	dHash
k-means	0.177182636398	0.145288694464
Ward	0.263924865617	0.269275549233

Table 5: Words as hashes of quarter-frames and entire frame. Scaling and cropping. 590 videos generated with 200 clusters. $(P, S) = (400, 10), (M, N) = (8, 8)$.

	aHash	dHash
k-means	0.115417521898	0.0922214172648
Ward	0.15361344235	0.12703564114

Table 6: Word as hash of entire frame. Scaling and cropping. 778 videos generated with 200 clusters. $(P, S) = (400, 10), (M, N) = (8, 8)$.

Similarly, mirroring (flipping frames over the vertical axis) benefits from the clustering domain by leveraging information from other videos. Table 7 shows the result of mirroring with the established quarter-frame + entire-frame hashing, while Table 8 shows it with only one video generated per seed to eliminate extra information (and so videos cannot piggyback on the correct cluster assignment of others).

	aHash	dHash
k-means	0.199281033407	0.165812525378
Ward	0.270517397739	0.238243007109

Table 7: Scaling and mirroring. 692 videos generated with 200 clusters. $(P, S) = (400, 10), (M, N) = (8, 8)$.

Finally, each frame is generated as random noise (static) to verify results. AMI for each scheme is close to 0.0, reflecting random assignments (Table 9).

	aHash	dHash
k-means	0.0505948787497	0.061612110758
Ward	0.103380624214	0.0644192877081

Table 8: Scaling and mirroring. 400 videos generated with 200 clusters. $(P, S) = (400, 10), (M, N) = (8, 8)$.

	aHash	dHash
k-means	-0.0151775401177	-0.0183229972316
Ward	-0.00288944285421	0.00013592377066

Table 9: Random noise (static). 705 videos generated with 200 clusters. $(P, S) = (400, 10), (M, N) = (8, 8)$.

Conclusion and Future Work

The results from this project show that video clustering can utilize the bag-of-words model effectively. Future work will seek to boost the effectiveness, examining the following:

- Multiple runs of k-means with different centroid seeds
- Estimating k (number of classes/clusters) using heuristics
- Perception of transformations in distorting or preserving content
- More robust features and extension to ontologies
- Multiclass labeling

References

Cheung, S.-C., and Zakhor, A. 2005. Fast similarity search and clustering of video sequences on the world-wide-web. *Multimedia, IEEE Transactions on* 7(3):524–537.

Jiang, Y.-G.; Ngo, C.-W.; and Yang, J. 2007. Towards optimal bag-of-features for object categorization and semantic video retrieval. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, 494–501. ACM.

Zhou, X.; Zhuang, X.; Yan, S.; Chang, S.-F.; Hasegawa-Johnson, M.; and Huang, T. S. 2008. Sift-bag kernel for video event analysis. In *Proceedings of the 16th ACM international conference on Multimedia*, 229–238. ACM.