OPTIMIZATION OF K-MEANS CLUSTERING FOR HUMAN ACTION CLASSIFICATION

Swapnil Vekhande
Dept. of Electrical and Computer Engineering
Virginia Tech
Blacksburg, VA 24060, USA
svekhand@vt.edu

Ashrarul Haq Sifat
Dept. of Electrical and Computer Engineering
Virginia Tech
Blacksburg, VA 24060, USA
ashrar7@vt.edu

Abstract— **This paper presents a novel approach to optimize running time of classification of human action using optimized k-means method. The optimized algorithm avoids unnecessary distance calculations when the distance between point and center is too small or too large. This is achieved by applying the triangle inequality in two different ways, and by keeping track of lower and upper bounds for distances between points and centers. First, Spatio-Temporal Interest Point (STIP) features are extracted from the given action sequence and classified into a set of labeled words using accelerated k-means clustering. Non-linear multi-channel Gaussian Kernel is used for classification. The proposed method promises to yield better real-time capabilities of human actions like ”handshake”, ”hug”, ”run”, ”sit down”, ”eat” etc.**

I. INTRODUCTION

Human action classification from video has become very important for smart surveillance, human-computer interaction, robotics and a lot of other applications. The research in this areas has also come a long way with various techniques in Machine Learning and Computer Vision. Realistic human action tend to be more complex to recognize. Thus, it remains an active are of research. Another interesting aspect is the processing time taken by these algorithms to process video files since there are lots of features that are extracted from a short video clip.

Relation to prior work:

In the context of object recognition in static images, these problems are surprisingly well handled by a bag-of-features representation [15] combined with state-of-the-art machine learning techniques like Support Vector Machines. Similarly, many papers have also been published on speeding up k-means or k nearest neighbor search using inequalities that are specific for Euclidean distance. One such algorithm was proposed by Wu[16] using kick-out condition.

II. MOTIVATION FOR THIS WORK

Automatic human action classification is of significant use in many applications, such as intelligent surveillance, content-based video retrieval and human-computer interaction. In robotics it is required to understand the intention and motivation of actions in the surrounding environment. The problem has been researched for years, but remains a very challenging task. spatial-temporal feature based approaches (e.g. [2, 4]) have been widely used in human action analysis and achieved promising results. The local features are regarded as visual words, then each action is described as a single descriptor using bag-of-words (BoWs) model. Although BoWs model is popular, it has an essential drawback of only focusing on the number of words but ignoring the spatial-temporal information. A novel approach has been proposed in [13] which takes spatio-temporal information into account. They have used k-means clustering method which could be replaced by optimized k-means clustering for better real-time results.

There has been numerous efforts to speed up k-means especially for larger datasets. The basic algorithm used most commonly today where centers are recomputed once after each pass through the data by Lloyd[1]. However, that paper only discusses quantization for some special one-dimensional cases. Kanungo[5] proposed an efficient modification to k-means but the most obvious source of inefficiency in the algorithm is that it passes no information from one stage to the next. When clusters exceed 10, it suffers heavily. On the other hand Elkan[14] proposed a novel acceleration which reduces redundant distance computation

III. PROBLEM FORMULATION

The k-means clustering used in the Laptev et. al.[1] method is the standard method. We propose to use accelerated k-means method which yields better real-time results for the BoVW method. Thus, the aim of this research is to optimize the algorithm in [13] with [14].

A. Characteristics of the problems

The issue of fast and accurate classification of video is still an open research problem. The optimization in this paper refers to time. This is so because human actions are particularly complex in that they could be misclassified into similar actions. There are lots of videos on the Internet which require sophisticated retrieval methods.Movies contain a rich variety and a large number of realistic human actions. Common action classes such as eating, walking and fighting are challenging for recognition.

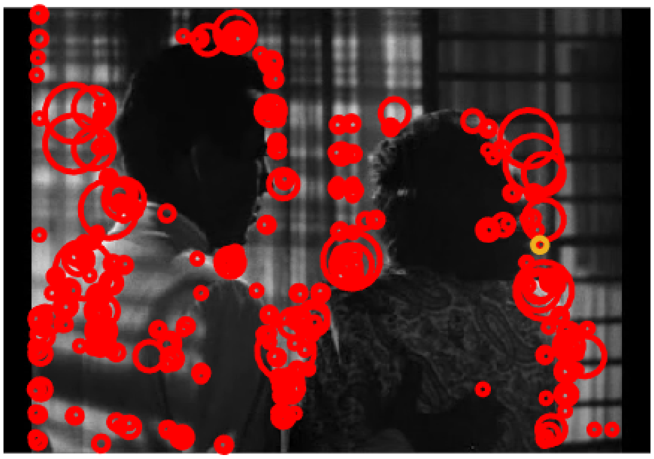

Fig. 1. Generated STIPs in the hug action.

B. Extraction of Space-time features

In this method videos are considered as volume of pixels. Spatio-temporal features are located at spatio-temporal salient points that are extracted using interest point operator like Harris detector in three dimensions. These points are also viewed as extension of 2D interest points with temporal information. Interest points detected for two frames with human actions are illustrated in figure 1. To characterize motion and appearance of local features, one needs to compute histogram descriptors of space-time volumes in the neighborhood of detected points. the Harris operator. We use a multi-scale approach and extract features at multiple levels of spatio-temporal scales here. The size of each volume (Δx, Δy, Δz) is related to the detection scales by $\delta, \Delta = 2k\sigma$, $\Delta t = 2kT$. Each volume is subdivided into a (nx, ny, nt) grid of cuboids; for each cuboid we compute coarse histograms of oriented gradient (HoG) and optic flow (HoF). Normalized histograms are concatenated into HoG and HoF descriptor vectors and are similar in spirit to the well known SIFT descriptor. The parameter values are $k = 9$, $nx, ny = 3$, $nt = 2$.

C. Spatio-temporal bag-of-features

The Bag-of-words model is mainly used as a tool of feature generation. Along similar lines here there is spatio-temporal BOF implies unstructured global representation of videos which is built using a large set of local features. This requires the construction of a visual vocabulary. In the following experiments we cluster a subset of 100k features sampled from the training videos with the standard k-means algorithm as well as accelerated k-means. The number of clusters is set to k = 4000, which has shown empirically to give good results and is consistent with the values used for static image classification. The BoF representation then assigns each feature to the closest (we use Euclidean distance) vocabulary word and computes the histogram of visual word occurrences over a space-time volume corresponding either to the entire video sequence or subsequences defined by a spatio-temporal grid. If there are several subsequences the different histograms are concatenated into one vector and then normalized.

D. Non-linear Support Vector Machines

Multi-channel Gaussian kernel has been used for classification.

$$K(H_i,H_j) = exp(-\sum_{c\in C}\frac{1}{A_c}D_c(H_i,H_j)) \qquad (1)$$

We use the same kernel as proposed by Zhang [17] for non-linear classification.

$$D_c(H_i,H_j) = \frac{1}{2}\sum_{n=1}^{V}\frac{(h_{in}-h_{jn})^2}{h_{in}+hjn} \qquad (2)$$

IV. Proposed Solution

As discussed in the problem formulation, our goal is to improve the real-time performance of the classification process by using an optimized version of K-means clustering. In this regard, there is standalone work in literature with the objective of optimizing the K-means method. We aim to incorporate the optimized method in this human action classification aspect.
To indicate the overall process, the total algorithm including the proposed improvement step is depicted in Algorithm 1. At first, the action is represented by H from video $\{I_t\}_{t=1}^{F}$ with F frames. Then algorithm 1 with modified k-means is run.

A. General K-means Algorithm

K-means[8] is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. K-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining.
To put it in terms of objective functions, the k-means uses Euclidean distance calculations and tries to minimize it for the nearest cluster. The objective function is given by,

$$J(C,X) = \sum_{x\in X} min_{c\in C}||x-c||^2 \qquad (3)$$

Here , $x \in \mathbf{X}$ is the data points and $c \in \mathbf{C}$ are the cluster centers. The goal is to find C that minimizes $J(C,X)$. The cluster centers have to be initialized with some initial values or there is a k-means++ algorithm that uses another algorithm to generate the initial k centers. Typically without that, random points are chosen for the initial centers.

It is considered fast as one need not compute exact distance between any two points. However, k-means is still slow for larger datasets.

B. Optimized K-Means Method

K-Means is based on the minimization of the average squared Euclidean distance between the data items and the clusters center (called centroid). The clusters center is defined as the mean of the items in a cluster. The standard implementation of K-Means consists of successive iterations.

Each iteration requires calculating distance of each point from centroid for each cluster.

Accelerated k-means is based on the fact that if the point is too far away or too close to centroid the actual distance need not be computed. Therefore, we need the accelerated algorithm to satisfy three properties. First, it should be able to start with any initial centers, so that all existing initialization methods can continue to be used. Second, given the same initial centers, it should always produce exactly the same final centers as the standard algorithm. Third, it should be able to use any black-box distance metric, so it should not rely for example on optimizations specific to Euclidean distance.

C. Objective function

The approach we derive to optimize k-means utilizes the triangular inequality constraint. That is for any three points x, y and z ,

$$d(x,z) \leq d(x,y) + d(y,z) \tag{4}$$

where d denotes the distance between respective points. The difficulty in this constraint is that we need to derive lower bounds and not the upper bound for the constraint condition and this one gives the upper bounds.

Now lets assume that x is any data point and c is assumed to be the center to which x is currently assigned. Let also that c' is any other center. Now according to [14], if $\frac{1}{2}d(c,c') \geq d(x,c)$ then $d(x,c') \geq d(x,c)$. Therefore, in such a case it would not be necessary to calculate $d(x,c')$. Now, if we do not know $d(x,c)$ exactly but we have an upper bound u such that $u \geq d(x,c)$, then we need to need to compute $d(x,c')$ and $d(x,c)$ only if $u > \frac{1}{2}d(x,c')$.

If $u \leq \frac{1}{2} mind(c,c')$ where the minimum is all over $c' \neq c$, then the point x must remain assigned to the center c, and all distance calculations for x can be avoided.

Now, Let, x is any data point and b is assumed to be the center to which x is currently assigned. Also b' is the previous version of the same center and in the previous iteration, we know that a lower bound l' existed such that $d(x,b') \geq l'$. It assumes that the centers are numbered 1 through k, and b is center number j, then b' is center number j in the previous iteration. Now, following the theory in [14], we can infer that a lower bound l exists for the current iteration by,

$$d(x,b) \geq max\{0, d(x,b') - d(b,b')\} \tag{5}$$

$$\geq max\{0, l' - d(b,b')\} = l \tag{6}$$

This is the objective function we try to optimize. Now, if l' is good approximation to the previous distance between x and the j^{th} center, and this center has moved only a small distance, then l is a good approximation to the updated distance.

The algorithm below is the one of the first k-means variants that uses lower bounds carries over varying information from one k-means iteration to the next. We assume that $u(x) \geq d(x,c)$ is an upper bound on the distance between x and the center c to which x is currently assigned, and also $l(x,c') \leq d(x,c')$ is a lower bound on the distance between x and some other center c'. If $u(x) \leq l(x,c')$ then $d(x,c) \leq u(x) \leq l(x,c') \leq d(x,c')$, so it is necessary to calculate neither $d(x,c)$ nor $d(x,c')$. it is noteworthy that it will never be necessary in this iteration of the accelerated method to compute $d(x,c')$, but it may be necessary to compute $d(x,c)$ exactly because of some other center c'' for which $u(x) \leq l(x,c'')$ is not true.

Now we describe the accelerated k-means algorithm.

Algorithm 1 Optimized K-Means algorithm

```
Pick initial centers
Set the lower bound $l(x,c) = 0$ for each point $x$ and center $c$
Assign each $x$ to its closest initial center $c(x) = argmin_c d(x,c)$
Each time d(x,c) is computed, set $l(x,c) = d(x,c)$. Assign upper bounds $u(x) = min_c d(x.c)$.
for centers $c$ and $c'$ do
    compute $d(c,c')$
    for all centers $c$ do
        compute $s(c) = \frac{1}{2} min_{c' \neq c} d(c,c')$
    end for
end for
Identify all points $x$ such that , $u(x) \leq s(c(x))$
for all remaining points $x$ and centers $c$ such that $c \neq c(x)$ AND $u(x) > l(x,c)$ AND $u(x) > \frac{1}{2}d(c(x),c)$ do
    if $r(x)$ then compute $d(x,c(x))$ and assign $r(x) = false$
    else $d(x,c(x)) = u(x)$
    end if
    if $d(x,c(x)) > l(x,c)$ OR $d(x,c(x)) > \frac{1}{2}d(c(x),c)$ then Compute $d(x,c)$
        if d(x,c) ¡ d(x,c(x)) then assign $c(x) = c$
        end if
    end if
end for
for each center $c$ do
    Let $m(c)$ be the mean of the points assigned to $c$
end for
for each point $x$ and center $c$ do
    Assign $l(x(c)) = max\{l(x,c) - d(c,m(c)), 0\}$
end for
for each point $x$ do
    assign
        $u(x) = u(x) + d(m(c(x)), c(x))$
        $r(x) = true$
end for
Replace each center $c$ by $m(c)$
```

The fundamental reason why the algorithm above is effective in decreasing the number of distance calculations is that at the start of each iteration, the upper bounds $u(x)$ and the lower bounds $l(x,c)$ are tied for most points x and centers *c*. If these bounds are tied at the start of one iteration, the update bounds tend to be tied at the start of the next iteration. This is because the location of the most centers changes only slightly and therefore the bounds change only slightly.

V. Experimental Methodology

The proposed optimization algorithm will be evaluated on two challenging datasets: Hollywood This datasets are standardized for multiple human actions. It contains six types of human actions, namely driving, eating, kissing, sitting down, handshake, hugging, get out of car, running, sitting up, fighting, talking on phone, standing up, performed several times. The sequences were taken for four different scenarios: outdoors, outdoors with scale variation, outdoors with different clothes and indoors. We train an SVM classifier for each action as being present or not following the evaluation procedure. The performance is evaluated with the average precision (AP) of the precision-recall curve. We use the optimized combination of spatio-temporal grids from . Table 5 presents the AP values for the two training sets and for a random classifier referred to as chance AP. Experiments were conducted on a machine consisting of an Intel i7-5930K CPU at 3.50GHz (12 CPUs) clock speed, 32 GB RAM memory. It also has two Nvidia GeForce GTX 970 graphics card, with a shared memory of 20 GB for faster processing. The confusion matrices for the new method will be compared against the existing DPF method results.

A. Dataset

The dataset is taken from the standard Hollywood dataset as depicted in figure 2. There are a lot of complex actions in this dataset and we choose 12 of them for our testing purpose.

B. Applying Optimized k-means

We applied normal k-means and optimized k-means clustering algorithm in the clustering step as described in section 2. For the unoptimized motion of simple walking and complex walking we recorded the processing time to be 0.436687 sec and 2.695634 sec. The characteristics of the two datasets is that one of them is walking in almost 2D plane with respect to the camera view and the complex motion is walking in the 3d plane with respect to the camera view. Therefore, the complex motion generated a lot of STIPs more compared to the simple walking. After applying the optimized k-means, the run time of the code reduces to 0.278082 sec and 0.437298 sec. Thus we can see atleast 50% improvement in the run-time. It also verifies the theory behind our experimentation that the larger the dataset, the more efficient the optimization becomes as we can see that for the complex motion it has improved by almost 150%.

C. Classification of Actions

We evaluate the performance of our classifications in terms of the precision-recall curve that is shown in figure 3. Each of the actions generate a separate precision-recall curve for different actions. The definition of the precision and recall are given as below equation.

Fig. 2. The dataset tested for classifying actions in different labels. From the top row going from the left to right, the actions depicted in this figure are driving, eating, kissing, sitting down, handshake, hugging, get out of car, running, sitting up, fighting, talking on phone, standing up .

$$Precision = \frac{tp}{tp+fp} \tag{7}$$

$$Recall = \frac{tp}{tp+fn} \tag{8}$$

where tp = True Positives, fp = False Positives and fn = False Negatives. This is a typical performance review option for the SVMclassifier in Matlab. We also generate the average of the precision curves in figure 4 which gives up the same mean precision of 0.536 for both the algorithms. Therefore, it is also verified that we are not compromising with the accuracy of the classifier. Actually it is pretty intuitive to get the idea of this result also because the k-means clustering step is not influencing the classifier anyway unless it is generating the wrong cluster. It also verifies the universal nature of the optimized algorithm and that it can be applied in many processes where process time is of paramount importance.

VI. Concluding Remarks

Action classification will remain an active research topic and as such will require optimization. The proposed method in one such step. The results of classification based on BOF are encouraging. Future work could revolve around using other kernels like Hellinger kernel. It also remains to be tested if the same algorithm could be extended for other human actions. Authors believe that holistic approach to classify both object and human action will go a long way in solving the problem. Also, the application of multimodal neural networks for the robust classification could also help in such problems.

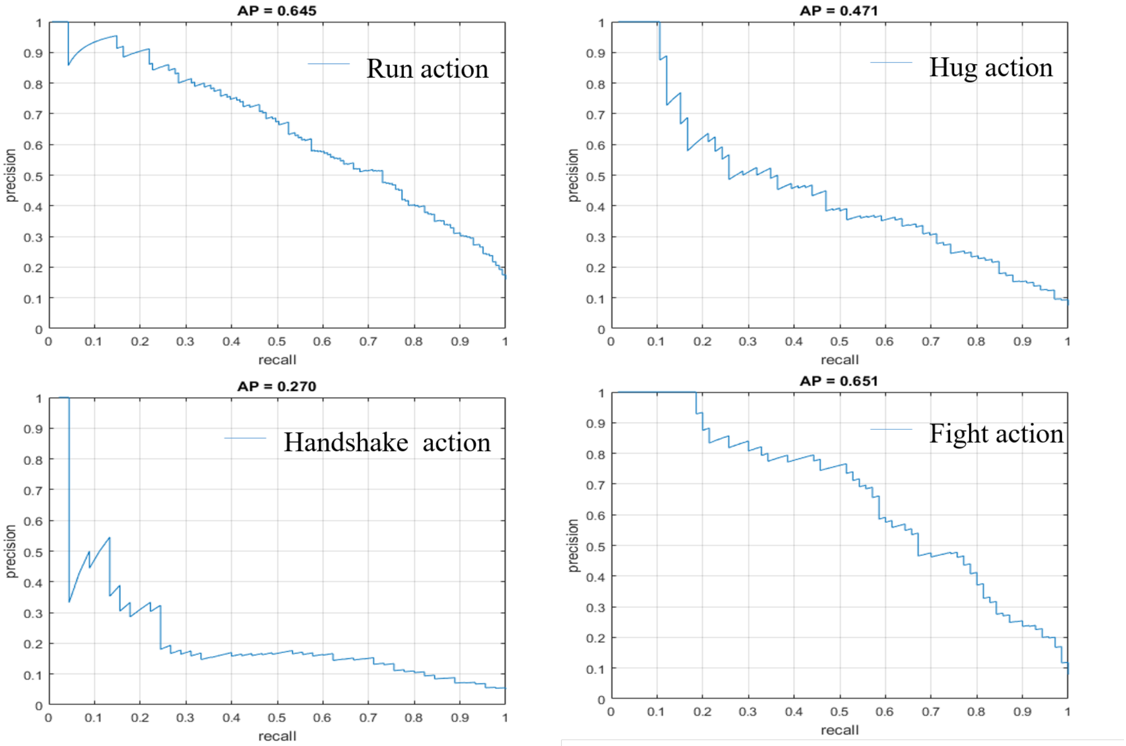

Fig. 3. Precision recall curves for different kinds of human actions from the Hollywood2 dataset.

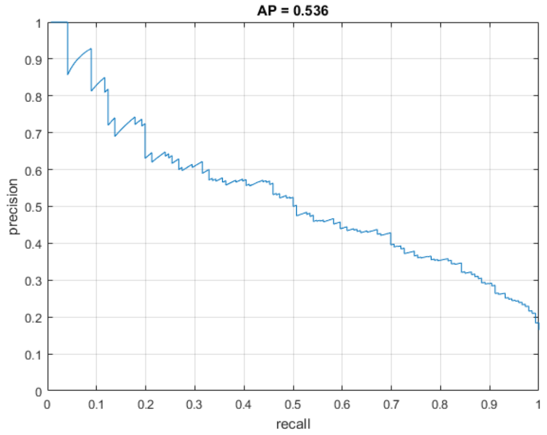

Fig. 4. Mean of all the action classification precision-recall from the actions specified in figure 2.

ACKNOWLEDGMENT

We would like to thank Dr. Yu for his encouraging lectures and feedback to our project which helped us a lot to organize our efforts.

REFERENCES

[1] Lloyd, S. P. (1982). Least squares quantization in PCM. IEEE Transactions on Information Theory, 28, 129137.

[2] Liu, H., Liu, M., & Sun, Q. (2014). Learning directional co-occurrence for human action classification. ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, 12351239. https://doi.org/10.1109/ICASSP.2014.6853794

[3] Nazir, S., Yousaf, M. H., & Velastin, S. A. (2018). Evaluating a bag-of-visual features approach using spatio-temporal features for action recognition. Computers and Electrical Engineering, 0, 110. https://doi.org/10.1016/j.compeleceng.2018.01.037

[4] Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., Wu, A. Y. (2000). The analysis of a simple k-means clustering algorithm. ACM Symposium on Computational Geometry (pp. 100109). ACM Press.

[5] Liu, M., Liu, H., Sun, Q., Zhang, T., & Ding, R. (2016). Salient pairwise spatio-temporal interest points for real-time activity recognition. CAAI Transactions on Intelligence Technology, 1(1), 1429. https://doi.org/10.1016/j.trit.2016.03.001

[6] Sun, Q., & Liu, H. (2012). Action Disambiguation Analysis Using Normalized Google-Like Distance. Accv2012, 425437. Engineering Lab on Intelligent Perception for Internet Of Things(ELIP), Key Laboratory for Machine Perception,Shenzhen Graduate School, Peking University, China

[7] L. Cao, Z. Liu, T.S. Huang, Cross-dataset action detection, in: CVPR, 2010, pp. 1998e2005.

[8] MacQueen, "Some methods for classification and analysis of multivariate observation" 1967

[9] Poteras, C. M., & Mih, M. C. (2014). An Optimized Version of the K-Means Clustering Algorithm, 2, 695699. https://doi.org/10.15439/2014F258

[10] R. Messing, C. Pal, and H. Kautz, Activity recognition using the velocity histories of tracked keypoints, in ICCV, pp.104-111, 2009.

[11] M. S. Ryoo, C. C. Chen, J. K. Aggarwal, and et al., An overview of contest on semantic description of human activities (sdha) 2010, in Recognizing Patterns in Signals, Speech, Images and Videos, pp.270-285, 2010.

[12] J. M. Carmona and E. J. Fernndez-Caballero, A survey of video datasets for human action and activity recognition, in CVIU, vol.117, Issue 6, pp.633-659, 2013.

[13] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, "Learning realistic human actions from movies", IEEE Conf. on Comp. Vision and Pattern Recogn, pp. 1-8, 2008.

[14] C. Elkan, "Using the Triangle Inequality to Accelerate k-means," in Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003).

[15] J. Willamowski, D. Arregui, G. Csurka, C. R. Dance, and L. Fan. Categorizing nine visual classes using local appearance descriptors. In IWLAVS, 2004.

[16] Wu, K.-S., Lin, J.-C. (2000). Fast VQ encoding by an efficient kick-out condition. IEEE Transactions on Circuits and Systems for Video Technology, 10, 5962.

[17] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. ”Local features and kernels for classification of texture and object categories: A comprehensive study.” IJCV, 73(2):213238, 2007.