

Activity classification using joint data - A Case study

Abishek Kumar
Drexel University
Dept. of Digital Media
Philadelphia, PA
ask85@drexel.edu

Prateek Goel
Drexel University
Dept. of Information Sciences
Philadelphia, PA
pg427@drexel.edu

Abstract—This project is an extension of the work done by Bearman et. al [1]. The work done in this project takes into consideration joint data from a large annotated activity data set and determines activities. The final product is a case study of various classification algorithms used to classify activities using the annotated joint data and a report of associated accuracy metrics for each of the methods.

Index Terms—case study, activities, joints, classification

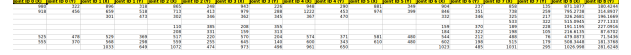
I. BACKGROUND

A. Activity classification

Using simple machine learning models and inbuilt libraries, various groups have combined the power of CNNs (Convolutional Neural Networks) and image data of Motion data sets to determine activities in the images [1]. Joint data is valuable in computing pose information as an estimation paradigm, as such pose information can be used to determine activities in input images. However determining human poses or the problem of human pose estimation is a non-trivial problem that involves identification of major body parts and joints. This step has been crucial in detecting activities, however the approach of this paper deals with avoiding the pose estimation problem and directly using the annotated joint data from an appropriate activity data set i.e. MPII human pose data set [2] to classify activities.

B. Joint data

The MPII human pose data set [2] is state of the art benchmark for human pose evaluation and activity classification with general categories such as carpentry, running, eating etc. but also contains specific activities under these categories such as motor scooter, ice skating, wind surfing etc. **The project aims to only classify the general categories.** The data set includes around 25K images containing over 40K people with annotated body joints. The images were systematically collected using an established taxonomy of every day human activities. Overall the data set covers 410 human activities and each image is provided with an activity label. Each image was extracted from a YouTube video and provided with preceding and following un-annotated frames. In addition, for the test set we obtained richer annotations including body part occlusions and 3D torso and head orientations.



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----

Fig. 1. Excel Data Snippet

II. RELATED WORK

Cipitelli et. al [3] had previously created an activity classifier using RGBD-sensors to aid assisted living individuals and their care takers to determine what the patient is doing at any given time. They used an SVM to determine multiclass activities from skeleton data acquisition from RGBD-sensors. Bearman et. al [1] produced a regression classifier using a CNN for pose estimation on the LEEDS sports data set and then used resulting human poses as inputs to another CNN classifier that determined the appropriate activities in the MPII data set.

III. METHODOLOGY

The methods in this paper will include a variety of classification algorithms to test the joint data set on. Firstly the joint data needs to be preprocessed for the entire MPII human pose data set. The joint data needs to be derived as an excel table wherein there are 2D coordinates for each of the 15 joints in the data set for each activity. The training set after preprocessing contains sheets with 2D joint data and the activity labels for each image in the data set. The methods described determine if knowing annotated joint data would improve the accuracy of classification algorithms.

A. Creating a JSON file

The MPII data set is first obtained as a .mat file and converted into a JSON file to be more human readable. However the JSON file contains a lot of garbage data that needs to be removed, during this process the activity label and 2D joint coordinates for each of the joints in an image as derived. The output of this method is the prior-mentioned excel data sheet.

B. Algorithms

All algorithms are implemented with standard library functions. The joint data set is classified using the following in the case study.

```

function DTL(examples, attributes, default) returns a decision tree
if examples is empty then return default
else if all examples have the same classification then return the classification
else if attributes is empty then return MODE(examples)
else
    best ← CHOOSE-ATTRIBUTE(attributes, examples)
    tree ← a new decision tree with root test best
    for each value  $v_i$  of best do
        examplesi ← {elements of examples with best =  $v_i$ }
        subtree ← DTL(examplesi, attributes – best, MODE(examples))
        add a branch to tree with label  $v_i$  and subtree subtree
    return tree

```

Fig. 2. Decision Tree algorithm

- Decision tree classifier: are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. The cost of using the tree (i.e., predicting data) is logarithmic in the number of data points used to train the tree.
- Support vector machine classifier: is capable of performing multi-class classification on a data set. They implement the “one-against-one” approach (Knerr et al., 1990) for multi- class classification. The aim is to maximize distance to closest example (of each type).
- K-nearest neighbors classifier: is a type of instance-based learning or non-generalizing learning: it does not attempt to construct a general internal model, but simply stores instances of the training data. Classification is computed from a simple majority vote of the nearest neighbors of each point: a query point is assigned the data class which has the most representatives within the nearest neighbors of the point.

```

k-Nearest Neighbor
Classify (X, Y, x) // X: training data, Y: class labels of X, x: unknown sample
for i = 1 to m do
    Compute distance  $d(X_i, x)$ 
end for
Compute set I containing indices for the k smallest distances  $d(X_i, x)$ .
return majority label for { $Y_i$  where  $i \in I$ }

```

Fig. 3. K-nearest neighbors algorithm

- Naive Bayes Classifier: is a probabilistic classifier that makes classifications using the Maximum A Posteriori decision rule in a Bayesian setting. It can also be represented using a very simple Bayesian network.

IV. RESULTS

A. Accuracy metric table

The following table shows the various accuracies for the algorithms implemented in this case study.

B. Confusion Matrices

The confusion matrices below show the correct number of predictions made with KNN and Decision Tree classifiers (Fig 4 and Fig 5). From the results it can be seen, in both KNN

TABLE I
ACCURACY TABLE

Table	Accuracy
KNN	0.294
Decision Tree	0.239
SVM	0.162
Naive Bayes	0.07

[4	5	1	2	2	0	0	5	0	0	3	0	0	0	11	0	0	0	2	6]
[4	79	2	4	12	5	0	7	4	2	6	0	1	0	20	0	0	0	5	3]
[0	2	2	0	0	1	0	1	0	1	2	0	0	0	6	0	0	0	0	1]
[0	3	4	10	8	5	0	9	0	3	10	0	0	0	6	0	0	0	1	1]
[1	8	3	8	51	6	0	12	6	4	19	1	1	0	10	0	0	0	1	2]
[5	6	1	6	18	28	0	11	5	2	24	0	0	0	9	0	0	1	0	3]
[1	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	2	0]
[6	8	2	6	11	4	0	28	1	1	18	0	0	0	23	0	0	0	2	3]
[0	4	0	5	9	3	0	1	13	3	14	0	0	0	5	0	0	0	0	1]
[0	1	1	2	7	2	0	0	1	13	4	0	0	1	2	0	0	0	0	0]
[4	21	3	3	25	14	0	10	5	4	62	0	1	1	26	0	0	2	3	5]
[0	2	0	3	0	0	0	0	1	0	2	0	0	0	0	0	0	0	0	0]
[1	0	2	0	2	0	0	4	0	0	2	0	3	0	5	0	0	2	0	1]
[0	1	0	0	3	1	0	0	0	0	1	0	0	7	0	0	0	0	0	0]
[15	18	6	12	15	9	1	22	1	3	20	1	1	1	84	0	0	0	10	8]
[0	0	0	0	1	0	0	0	0	0	3	0	0	0	1	1	0	0	0	0]
[0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0]
[2	4	1	2	4	4	0	4	1	0	7	0	0	0	10	0	0	1	0	0]
[3	11	0	2	3	3	0	8	1	1	11	0	0	0	25	0	0	0	19	6]
[4	5	4	2	4	1	0	4	1	2	6	0	1	0	19	0	0	0	3	6]

Fig. 4. Confusion matrix - KNN

and Decision Trees, the number of correct predictions for any activity is clearly higher than the rest. Inspite of having incorrect classifications as well, the number of correct activity classifications based only on joint data are definitely higher (61 to 70 out of 100 per activity).

V. CONCLUSION AND LIMITATIONS

Based on the results displayed, we can safely conclude that joint locations have an impact when classifying images based on the activities. As can be seen by the confusion matrices, the number of correct predictions are the highest against the incorrect predictions. So, although the accuracy of the entire model is low, it gives enough information to place reliance on the idea that joint data can indeed prove useful.

During the course of our project, we came across multiple obstacles that have caused us to change our direction. First,

[4	0	0	2	3	2	0	1	0	1	6	0	0	0	10	0	0	3	3	6]
[2	61	1	7	9	6	0	12	6	3	11	0	1	0	14	2	0	0	14	5]
[0	2	1	1	2	1	0	1	0	0	1	0	0	0	6	0	0	1	0	0]
[3	2	0	9	7	3	0	3	1	3	11	0	1	0	12	0	0	0	3	2]
[2	3	0	9	35	7	0	9	2	1	33	3	0	1	14	0	0	1	7	6]
[2	12	0	4	11	23	0	4	10	4	27	0	0	0	14	0	0	0	5	3]
[0	1	0	0	0	1	0	0	0	1	2	0	0	0	0	0	0	0	2	0]
[3	9	0	3	11	6	0	25	0	2	24	0	0	0	17	0	0	1	4	8]
[1	3	0	2	4	4	0	1	9	4	16	1	0	0	8	0	0	1	3	1]
[1	3	0	1	3	3	0	0	2	9	7	0	0	0	3	0	0	1	0	1]
[3	13	1	13	29	18	0	14	6	5	48	0	0	0	24	0	0	1	10	4]
[0	0	0	3	0	0	0	0	2	0	2	1	0	0	0	0	0	0	0	0]
[1	1	1	1	0	3	0	3	0	0	4	0	1	0	6	0	0	0	1	0]
[0	1	0	0	0	1	0	0	1	1	6	0	0	0	3	0	0	0	0	0]
[10	28	4	8	16	16	0	17	0	4	25	0	1	1	70	0	0	2	13	12]
[0	0	0	0	0	0	0	0	1	0	0	0	0	0	3	0	0	2	0	0]
[0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0]
[1	2	0	2	4	4	0	4	0	0	7	0	0	0	12	0	0	1	1	2]
[1	10	1	5	3	4	0	6	1	1	9	1	0	0	24	0	0	1	21	5]
[0	5	2	4	1	2	0	1	1	0	12	0	0	0	15	0	0	0	5	14]

Fig. 5. Confusion Matrix - Decision Tree algorithm

due to time constraints and lack of server space we were forced to alter our approach and thus decided to use existing low accuracy models to begin our study (KNNs, Decision Trees etc.) as opposed to CNNs as discussed originally. Furthermore, for same reasons we were forced to use just joint annotations as opposed to superimposing joint data with image data. Nevertheless, using just the joint annotations, our study still provides enough results to show us moving in the right direction.

REFERENCES

- [1] Bearman, Amy, and Catherine Dong. "Human pose estimation and activity classification using convolutional neural networks." CS231n Course Project Reports (2015).
- [2] MPII human pose dataset.
- [3] Cipitelli, Enea, et al. "A human activity recognition system using skeleton data from RGBD sensors." Computational intelligence and neuroscience 2016 (2016).