Assignment 1 - Arrows Recognition

Li-Hen Chen 928003907

Department of Computer Science & Engineering, Texas A&M University

Abstract— This project was developed as partial fulfillment of the requirements for the course CSCE 624, Sketch Recognition taught by Dr. Tracy Hammond during the Fall 2019 at Texas A&M University. This project uses sketch features to recognize arrows as opposed to other shapes. The main motto of this project is feature engineering for better recognition of arrows versus other shapes. Achieved a whopping 99.56% accuracy using 10-fold cross-validation and Random-Forest classifier on WEK

1. Problem Description

This project uses sketch features to recognize arrows as opposed to other shapes. The main motto of this project is feature engineering for better recognition of arrows versus other shapes.

Initially we have some strokes which are composed of (x, y, time) so we can use the information to identify our own features. Then, we also create new and more efficient features for better recognition. Final step was to extract the results to a csv which can be loaded into WEKA software for application of various machine learning models for the actual recognition. ZeroR, J48, Random Forest, and one other classifier of our choice was permitted.

2. Features Description

Feature 1 – 13 are the features from Rubine's paper. Feature 14 – 23 are features from Long's paper. Feature 24 – 30 are features coming from myself.

2.1 Features Explanation

1) Cosine of Initial Angle

The angle between the first and the third point is measured and angle's cosine is used as features rather than the angle itself to avoid a discontinuity as the angle passes 2pi and wraps to 0.

$$\frac{x_2 - x_0}{\sqrt{(x_2 - x_0)^2 + (y_2 - y_0)^2}}$$

This makes sure that shapes with similar shapes have similar values. This proved to be one of the moderate features for this task. Although this may take a great deal of importance in the other sketch recognition tasks, for arrow recognition, it can take any values as arrows can be at any directions, hence this might be the actual justification why this feature didn't prove to be one of the important features.

2) Sine of Initial Angle

The angle between first and third point is measured and angle's sine is used as features rather than the angle itself to avoid a discontinuity as the angle passes 2pi and wraps to 0.

$$\frac{y_2 - y_0}{\sqrt{(x_2 - x_0)^2 + (y_2 - y_0)^2}}$$

Similar argument goes for this feature as well. This makes sure that shapes with similar shapes have similar values. This feature too was moderate. For arrow recognition, it can take any values as arrows can be at any directions, hence this might be the actual justification why this feature didn't prove to be one of the important features.

3) Length of Bounding Box

This proved to be an important feature in this task. The diagonal length of bounding box gave one of the highest single accuracies of 95.67 % alone, which is pretty strong for

this task. The reason why I have kept this in my final sub-set is the same. For non-arrow shapes, this feature might be same for some cases, but looking at the dataset, it differed.

4) Angle of Bounding Box

Similar argument goes for this feature as well, but this was more than moderate feature for this task. As the arrows are facing in any direction, this feature value came out to be varying too much with the change of arrow direction. Still giving a good accuracy made this feature a feature in the final subset.

5) Distance between the first and the last point (Diagonal length)

This distance actually varies with user to user, that's why its performance was moderate. Some users tend to draw arrows in other fashion than the others so their starting and ending points are not representative of the shape. The others also had similar kind of distances between their initial and final points. Hence this feature might give out moderate results.

6) The Cosine between the first and last point

Such features hugely depend on which direction the arrow is facing, hence again a moderate feature in my opinion. As the arrows are facing in any direction, this feature value came out to be varying too much with the change of arrow direction. Still giving a good accuracy made this feature a feature in the final subset.

7) The Sine between the first and last point

Similar argument goes for this feature as well. Such features hugely depend on which direction the arrow is facing, hence again a moderate feature in my opinion. This makes sure that shapes with similar shapes have similar values.

8) Total Gesture Length

This is an important feature for this task. That's because even though this feature, by itself is not that great (as it depends on the user who is sketching), but this feature is used to normalize various metrics to remove this "user bias".

9) Total Angle traversed

This has also proved to be an important feature by itself, giving a high accuracy when only this feature is used. For arrows, the total rotation lies in a range, even though they may point anywhere.

10) Sum of the absolute value of the angle at each mouse (Curviness)

This feature measures the actual rotation of the sketch. Arrows typically have this feature is a range, where as for other figures, this might vary on a huge basis.

11) Sum of the squared value of those angles (Sharpness)

This metric grows up as the number of corners grow up, almost in an exponential way. This makes the feature to be a distinguishing feature of arrows from all the other shapes.

12) Maximum speed of gesture

This is a misleading feature, as the time taken to draw a sketch is very much dependent on the user who is sketching.

13) The duration of the gesture

This feature depends on the user, someone draws slow while someone draws fast, it cannot be the primary feature the recognize the sketch.

14) Aspect

This is a Long feature and it diminishes the difference between arrows with starting angles less than 45 and arrows with starting angles greater than 45, which seems to be essential in this task.

15) Curviness

Curviness of a gesture was computed by adding up all inter segment angles within the gesture whose absolute value was below a threshold (19 °). The threshold was chosen so that

the metric would agree with the author's curviness judgements of gestures in trial 1.
This unique feature proved to be very important feature. The curviness of arrows is generally less than the other sketches given in this classification task which made this feature stand out on its own. This makes sure that shapes with similar shapes have similar values.

16) Total angle traversed / total length
The normalization of the Total angle feature by stroke length proves to be important as it makes this feature invariant to different stroke length arrows as opposed to other shapes.

17) Density metric1
The normalization of the initial to last points distance feature by stroke length proves to be important as it makes this feature invariant to different stroke length arrows as opposed to other shapes.

18) Density metric2
The normalization of the diagonal distance feature by stroke length proves to be important as it makes this feature invariant to different stroke length arrows as opposed to other shapes.

19) Non-subjective openness
This normalization did not provide much classification of arrows as opposed to other shapes.

20) Area of bounding box
The argument of Long that this feature is not that important as compared to others proved to be true in this task. Area also does not take the shape of the sketch into consideration, hence can be counted as a moderate feature.

21) Log(area)
This feature makes a little difference in the classification task. As the logarithm is taken, the overall effect of total area is diminished, which is desirable.

22) Total angle / total absolute angle
This measures the approx. total number of non- absolute rotations, which is close in some range for arrows as opposed to other shapes.

23) Log(total length)
For this classification task it was seen that the stroke length was important but not linearly, but logarithmically. Higher stroke length does not mean that this feature must be weighted linearly, but a logarithm of that feature proved to be giving good results.

24) Log(aspect)
This feature makes a little difference in the classification task. As the logarithm is taken, the overall effect of total area is diminished, which is desirable.

2.2 Features Motivation

25) Average speed
This is new feature created by myself. Although this depends on which user is sketching, but let’s compute to see if it matters.

26) Stroke length divided by initial point to centroid point
This is a new feature which proved eminent in this problem. The centroid can be defined as center of gravity of a particular symbol.

27) Diagonal Length divided by initial point to centroid point
This is a new feature which proved eminent in this problem. For arrows, this feature lies in a range and for other shapes, this feature is just random

28) Distance between initial point and centroid point
This feature is just the Denominator of previous the features. Although I don’t think it will work on sketch recognition, let’s see how it performs.

29) Velocity

This is a misleading feature, as the time taken to draw a sketch is very much dependent on the user who is sketching.

30) Number of sketches

This is the number of sketches detected by the machine. It's looks like no sense for the sketch recognition for human, but let's see whether it means something to the machine.

3. Weka Classifier

In this part, I would train the data by different models **with 30 attributes.** In the next part, I would use Weka subset selection to find the best subset and train the model with selected subset.

First, I would describe the different models I choose and list the results. The data is composed of 500 arrows and 500 non-arrows. Each classifier I choose an algorithm to train. I use 10-fold cross validation to verify the data. Why use 10-fold cross validation? The main reason is that the k-fold cross-validation estimator has a lower variance than a single hold-out set estimator, which can be very important if the amount of data available is limited. If you have a single hold out set, where 90% of data are used for training and 10% used for testing, the test set is very small, so there will be a lot of variation in the performance estimate for different samples of data, or for different partitions of the data to form training and test sets. k-fold validation reduces this variance by averaging over k different partitions, so the performance estimate is less sensitive to the partitioning of the data. You can go even further by repeated k-fold cross-validation, where the cross-validation is performed using different partitioning of the data to form k sub-sets, and then taking the average over that as well.

Tree ➔ Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

Accuracy	**Confusion Matrix**	Pos	Neg
99.3%	Pos	497	3
	Neg	4	496

Rules ➔ Decision Table

Decision tables are a concise visual representation for specifying which actions to perform depending on given conditions. They are algorithms whose output is a set of actions. The information expressed in decision tables could also be represented as decision trees or in a programming language as a series of if-then-else and switch-case statements.

Accuracy	**Confusion Matrix**	Pos	Neg
98.8%	Pos	498	2
	Neg	10	490

Bayes ➔ BayesNet

A Bayesian network, Bayes network, belief network, decision network, Bayes(ian) model or probabilistic directed acyclic graphical model is a probabilistic graphical model (a type of statistical model) that represents a set of variables and their conditional dependencies via a directed acyclic graph (DAG). Bayesian networks are ideal for taking an event that occurred and predicting the likelihood that any one of several possible known causes was the contributing factor. For example, a Bayesian network could represent the probabilistic relationships between diseases and symptoms. Given symptoms, the network can be used to compute the probabilities of the presence of various diseases.

Accuracy	**Confusion**	Pos	Neg

	Matrix		
99.0%	Pos	494	6
	Neg	4	496

Functions ➔ MultilayerPerceptron

A multilayer perceptron (MLP) is a class of feedforward artificial neural network. An MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable.

Accuracy	**Confusion Matrix**	Pos	Neg
95.3%	Pos	492	8
	Neg	39	461

4. Weka Subset selection

The algorithm for subset selection I chose was InforGainAttribute. The InfoGain class is an implementation of a feature selection method by information gain.Information gain (InfoGain(t)) measures the number of bits of information obtained for prediction of a class (c) by knowing the presence or absence of a term (t) in a document.Concisely, the information gain is a measure of the reduction in entropy of the class variable after the value for the feature is observed. The attributes are then ranked from highest to lowest merit. I used **top 5 (f17, f23, f8, f26, f15) and top 13 features (f17, f23, f8, f26, f15, f10, f18, f19, f21, f20, f22, f3, f27)** to test on the different models and saw how these features influenced the results.

Tree ➔ Random Forest

- 5 Features

Accuracy	Confusion Matrix	Pos	Neg
99.0%	Pos	497	3
	Neg	7	493

- 13 Features

Accuracy	Confusion Matrix	Pos	Neg
99.3%	Pos	497	3
	Neg	4	496

Rules ➔ Decision Table

- 5 Features

Accuracy	Confusion Matrix	Pos	Neg
98.7%	Pos	499	1
	Neg	12	488

- 13 Features

Accuracy	Confusion Matrix	Pos	Neg
99.1%	Pos	500	0
	Neg	9	491

Bayes ➔ BayesNet

- 5 Features

Accuracy	Confusion Matrix	Pos	Neg
98.5%	Pos	495	5
	Neg	10	490

- 13 Features

Accuracy	Confusion Matrix	Pos	Neg
98.4%	Pos	495	5

	Neg	11	489

Functions ➔ MultilayerPerceptron

- 5 Features

Accuracy	Confusion Matrix	Pos	Neg
98.9%	Pos	497	3
	Neg	8	492

- 1 Features

Accuracy	Confusion Matrix	Pos	Neg
95.3%	Pos	492	8
	Neg	39	461

5. Discussion

I also trained on the training set without cross validation and the performance was incredibly high. However, when I performed cross validation, the results were around about 99%. According to the table above, we can see that the model performed best was Random Forest, which can be applied to lots of tasks according to my experiences. After I did subset selection, the accuracy was almost the same. The reason may be these models already did subset selection inside the models when training the models so we don't have to do extra subset selection when we train a model. When we just used 5 features, the running time reduced a lot, the accuracy was still high enough compared we used 30 features. When we used 12 features, the running time was less than 30 features case, and the accuracy also remained pretty high.

To sum up, features selection can find the most useful features and further reduce the total running time when training models.